ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF TECHNOLOGY
DEPARTMENT OF CIVIL ENGINEERING

Strength Evaluation of Existing Bridges

A Thesis Submitted to School of Graduate Studies in Partial Fulfillment of the Requirement for the Degree of Master of Science in Civil Engineering

By: Yonnas Bayou

Advisor: Dr. Asnake Adamu

January 2009

Addis Ababa

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

STRENGTH EVALUATION OF EXISTING BRIDGES

YONNAS BAYOU

APPROVED BY BOARD OF EXAMINERS

Dr. Asnake Adamu	_______________	_________________
Advisor		Date
______________	_______________	_________________
Examiner		Date
______________	_______________	_________________
Chairman		Date

Acknowledgement

First of all I would like to thank my God, for all things are for Him and by Him. I also thank my advisor Dr. Asnake Adamu for his guidance and encouragement from the beginning to the end. His cooperativeness and friendly manners made every thing easy.

I would like to thank my dear wife Betsega Tadesse, whose love and moral support made life fun and attractive. Last but not least I would like to thank my family who has always been there for me.

Abstract

A bridge must be inspected on regular basis to ensure public safety and to protect public investment. Bridge inspection program helps to identify the needs of bridges for repair, maintenance, preservation, reconstruction and replacement. Many bridges have collapsed in Ethiopia and other countries. Periodic inspection and maintenance of bridges helps to avoid this problem.

Recently bridge inventory and inspection program has begun in Ethiopia. This program has proved to be very useful in providing information about bridges in Ethiopia which otherwise would have been unavailable. According to a report from this program, quite a large number of bridges are in critical condition needing immediate intervention. This can be disastrous if remedial actions are not taken by the concerned bodies in due times.

One of the key components of a bridge inspection is load rating. This is a method to know the strength of an existing bridge to carry current traffic loads. When the load rating of a bridge is greater than or equal to one, it implies that the bridge is strong enough for current traffic conditions. Load rating or strength evaluation of existing bridges is quite different in some respect from designing of a new bridge. This is because in strength evaluation, the loads are taken to represent the actual traffic condition.

Since bridge inspection and maintenance activities started recently in Ethiopia, much is not known about strength evaluation of bridges. This thesis has thus made an attempt to address such problem by developing a computer program that will be used to load and evaluate the rates of simple span slab type and girder concrete bridges.

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Background

Bridges are important structures in providing access to cross valley and/or river in a road transportation system of any country. It is known that Ethiopia has a large number of bridges and also requires further more as new roads are constructed. Most of the existing bridges were constructed during the Italian occupation, between the year 1935 and 1941 [2]. As a result most bridges are older than 60 years. Many damages to most bridges have occurred and are still to occur due to different reasons, while some bridges have even collapsed [1]. A number of factors can be attributed to such causes. To mention some:

i. Most bridges are very old as mentioned above.
ii. Old bridges were designed for smaller traffic load than the load they are currently exposed to.
iii. Some members of a bridge have been damaged by different accidents.
iv. Due to overlay of finishing materials additional dead loads are introduced.
v. Most bridges can be overloaded due to introduction of new vehicles and illegal traffic practices.

After a bridge is constructed and left open to traffic, it must be continuously evaluated and rated in order to ensure public safety. In addition to ensuring public safety, the rating provides useful information to repair, rehabilitate, post maximum load to traverse, or close and replace existing bridges. Bridge rating is a means to check the capacity of existing bridge to carry loads.

Following the tragic loss of 46 lives when Silver Bridge collapsed, US initiated bridge inspection program in order to ensure the safety of the public and protect public investments [10]. Likewise other nations acknowledged the necessity of periodic bridge inspection and maintenance programs. Currently the Ethiopian Roads Authority (ERA) and other responsible bodies are carrying inspections and inventory of bridges in the country in order to know their performance and come up with possible solutions.

1.2 Objective of the Thesis

The main objective of this thesis is to develop a computer program that will be used to evaluate the strength of existing bridges. The computer program is sought to increase the efficiency and accuracy of the computational effort. It is also the aim of the thesis to show the methods and procedures of bridge rating. Since design of new bridges is different from strength evaluation of existing bridges, much is not known about strength evaluation. Hence this thesis is useful in creating awareness of strength evaluation of bridges.

1.3 What is in the Thesis

- There are five chapters in this thesis. The first chapter is introduction.
- Bridge inspection and ratings are discussed in chapter two. In this chapter brief history of bridges both internationally and locally is discussed. International and Ethiopian practice of bridge inventory, inspection and ratings are also discussed in detail. Moreover, recommendations for bridge inspection and rating to be practiced are forwarded.
- Strength evaluation of existing bridges is presented in chapter three. Detail discussion of the chapter includes the following points.
    - What is bridge strength evaluation
    - Methods and procedures for bridge strength evaluation
    - Loads used for strength evaluation
    - Rating factor equation
- The computer program for strength evaluation of bridges is discussed in chapter four. This chapter presents how action effects and strengths for shear force and

moment are calculated. It also discusses how the computer program is developed. In order to show how the program works, illustrative example is provided.

- Discussion and conclusion are presented in chapter five.

2. Bridge Inspection and Ratings

2.1 History of Bridges

The first bridges were probably made by nature, as for example a fallen log crosses a stream. Primitive man must have built many crossings over shallow streams by piling rocks for piers and covering them with slabs of stone, logs, or falling trees so as to span small rivers. Primitive stone bridges were simple arch bridges developed from slate stones laid on top of each other with slight offset until they meet in the middle. This kind of arch bridges can be found in ancient Egypt, Greece, Asia Minor and Mexico [7].

The stone arch was the first major advance in bridge design. The Roman period was a peak in the history of masonry arch bridge construction. The Romans perfected arch design using arches to build massive stone bridges throughout the Roman Empire. Some of these bridges have survived the wear and tear of history and still stand today [13]. Stone arch construction remained the premier bridge design until the introduction of the steam locomotive in the early 19th century [8].

Between the year 1830 and 1880, bridge construction begun to assume a more scientific aspect than before. The production of iron in commercial scale gave new possibilities. During this period iron trusses replaced stone arches and wooden bridges as the preferred design for larger bridges. The invention of converting cast iron into steel revolutionized bridge building. In the late 19th century and the first half of the 20th century, many large scale steel suspension bridges were constructed over major water ways.

Also in the late 19th century, engineers began to experiment with concrete reinforced with steel bars for added strength. In the mid-to-late 20th century, the steel and concrete girder bridge was one of the most commonly used bridge design. The last three decades of the 20th century saw a period of large scale bridge building in Europe and Asia [8].

Bridge building in Ethiopia seems to have begun in the 17th century with the help of the Portuguese. The first stone masonry bridge was built in 1626 on the Blue Nile River [4]. Then in this period a number of other arch bridges were built.

In the 18th century and the first half of 19th century bridge construction almost ceased, and those built by the Portuguese were being destroyed by the warring clans of Ethiopia. But there were some isolated instances of bridge building like those built by Emperor Theodros, the British Expedition to Megdella and King Tekle Haymanot who imported craftsmen from Italy.

The period after 1882 marks the revival of bridge building in Ethiopia. Emperor Menelik II employed Italian, Indian, and Greeks for building many bridges [4]. Nevertheless, most of bridges in Ethiopia were built during the Italian occupation, 1935-1941.

2.2 Bridge Inventory and Inspection

2.2.1 General

Bridge inspections are conducted to determine the physical and functional conditions of a bridge and assist to determine the basis for the evaluation and take preventive and maintenance actions. Bridge inventory on the other hand deals with record of quantifiable item of bridges which may be carried once in a bridge's life span. However, inspection need be carried frequently as it deals with condition assessment.

In this view outcomes from inventory and inspection of bridges include:

- Load rating of bridges

- Analysis for overload permit applications
- Initiation of maintenance actions
- Provide a continuous record of bridge condition and rate of deterioration
- Establish priorities for repair and rehabilitation programs

For successful bridge inventory and inspection proper planning, development of techniques and procedures, adequate equipment, and the professional experience and reliability of the personnel performing the inspection are important.

The interval for bridge inspection should be regular and preferably not more than two years [2]. But this interval can be shorter or longer than two years depending on factors such as age, traffic volume, size, susceptibility to collision, extent of deterioration, performance history of the bridge type, load rating, location, national defense designation, detour length, and social and economic impacts due to bridges being out of service.

Personnel involved in the various bridge inspection activities must be qualified for their specialized jobs. In general, depending on the level of responsibility, they must be knowledgeable in the various aspects of bridge engineering including design, load rating, construction, rehabilitation, and maintenance.

The bridge inspection process is the foundation of the entire bridge management operation and the bridge management system. Information obtained during the inspection will be used for determining required maintenance and repairs, in order to prioritize for rehabilitation and/or replacements as per extent of damages and then allocating resources. Moreover, it also enables to evaluate and make improvements to design of new bridges.

2.2.2 Types of Bridge Inspection

Basically there are five different types of bridge inspections:

- Initial
- Routine
- Damage
- In-depth
- Special Inspections.

Depending on the intensity of the inspection required, inspection frequency and the type of structure and details, a bridge may be inspected by one of the category of inspection types. An inspection event, particularly for large, complex, or deficient structures, often requires that a variety of inspection types be performed [9].

A. Initial Inspections

An initial inspection is the first inspection of a new or existing structure, as it becomes part of the bridge inventory. And also, change in the configuration of the structure or change in bridge ownership may also require an Initial Inspection. The Initial Inspection is a fully documented investigation accompanied by analytical determination of load carrying capacity.

There are two purposes of Initial Inspection:

i. It is used to provide all Structure Inventory and Appraisal (SI&A) data.
ii. It is also used for the determination of baseline structural conditions and the identification and listing of any existing problems or locations in the structure that may latter warrant special attention.

Per [9], it is being noticed that unexpected problems with a small number of newly constructed bridges have demonstrated that safety inspections may be needed even for new bridges to ascertain their initial and long-term safety, hence indicating the necessity of bridge inspection for bridge constructed at any time.

B. Routine Inspections

Routine Inspections are regularly scheduled inspections consisting of observations and/or measurements needed to determine the physical and functional condition of the bridge, to indentify any changes from "Initial" or previously recorded conditions, and to ensure that the structure continues to satisfy present service requirements.

Routine Inspections serve to document sufficient field observations/measurements and load ratings needed to:

- Determine the physical and functional condition of the structure.
- Identify changes from the previously recorded conditions.
- Determine the need for establishing or revising a weight restriction on the bridge.
- Determine improvement and maintenance needs.
- Ensure that the structure continues to satisfy present service and safety requirements.
- Identifying and listing existing problems
- Identifying and listing concerns of future conditions
- Indentify any inventory changes from the previous inspection.

C. In-Depth Inspections

An In-Depth Inspection is a close-up, hands-on inspection of one or more members above or below the water level to identify any deficiencies not readily detectable using Routine Inspection procedure. It serves to collect and document data to a sufficient detail needed to ascertain the physical condition of a bridge. When necessary or appropriate, nondestructive field tests and/or other material tests may need to be performed. In-Depth Inspections can be conducted alone or as part of a Routine or other type of inspection.

D. Damage Inspections

Damage Inspections are performed to assess structural damage following extreme weather-related events, earthquakes, vandalism and vehicular/marine traffic crashes. The scope of inspection should be sufficient to determine the need for emergency load restrictions or closure of the bridge to traffic, and to assess the amount of work necessary to effect a repair.

E. Special Inspections

A Special Inspection is an inspection scheduled at the discretion of the Bridge Owner. It is used to monitor a particular known or suspected deficiency, such as foundation settlement, scour, member condition, and the public use of a load-posted bridge [2].

2.2.3 Inspection Forms and Reports

Bridge owners are to maintain complete, accurate, and up-to-date record for each of their bridges. The bridge inspection file is an integral part of an effective bridge inspection and management system. These records are needed to establish an inventory and infrastructure assets. The information in the bridge inspection file is kept current through bridge inspection cycle. Records containing the inventory and condition information for bridges and structures are a vital key to managing these critical assets and assuring public safety.

Inventory and inspection records should be prepared and maintained in accordance with the standard of the country. Structure Inventory & Appraisal (SI&A) data should be maintained for each bridge and updated after repairs, rehabilitations, or modification of the bridge. The accuracy and consistency of the inspection and documentation is vital because not only does it impact programming and funding appropriation but also it affects public safety.

Maintaining complete, accurate, and up-to-date record for each bridge helps to:

- Establish an inventory of infrastructure assets
- Document the condition and functionality of infrastructure, including the need and justification for bridge restrictions, for public safety
- Identify improvement and maintenance needs for planning and programming
- Document improvements and maintenance repairs performed
- Provide available information in a timely manner for safety inspection

The inspection file should include:

- List of bridges and structures
- List of posted bridges with date of most recent signing verification
- List of bridges with special features and/or conditions that necessitate special or more frequent inspections
- List of bridges that require underwater inspection
- List of bridges to be inspected during/after high water events
- Contact list for key staff during bridge emergencies
- Organizational Chart listing key staff, program managers and inspectors.
- Certification credentials for the program manager, inspectors and key staff.
- Quality Control Plan including findings and results
- Quality Assurance Reviews Plan including findings and results
- List of inspection equipment
- List of bridge design and inspections reference materials
- Pass results of Quality Assurance reviews

2.2.4 International Practice

2.2.4.1 American Practice

After World War II, U.S. initiated an extensive road construction program. As a result, most emphasis was on new and economical construction, and for about two decades most highway departments gave little effort to bridge inspection or preventive maintenance.

In 1967, there was a sudden collapse of the Silver Bridge, a pin-connected link suspension bridge over the Ohio River at Point Pleasant, West Virginia, with loss of 46 lives. As a result, a 1968 Federal act initiated a national bridge inspection program that recognized the need for periodic and consistent bridge inspections [11]. The National Bridge Inspection Standards (NBIS) were developed after the 1968 Federal Highway Act became effective [9]. The Bridge Inspection Program is federally mandated and has been in effect since 1971.

Annual inspection for the NBIS requires each state to update the state inventory within 90 days. All inspection must be uploaded into the Bridge Management System (BMS) no later than March 15th for the previous year's inspection. If it is a new bridge, the Structure Inventory & Appraisal (SI&A) data is also entered into the State BMS database inventory within 90 days after completion of the work.

Different states in America have a decentralized Bridge Inspection Program that follows the guidelines and standards established by the Federal Highway Administration (FHWA) and the American Association of State Highway and Transportation Officials (AASHTO) [9]. Each District Office manages and administers the inspection of department bridges in its area. The central office controls that each District Office complies with Federal directives. This includes making sure that all structures are inspected at the proper interval and that the state structure files are kept up-to-date and accurate. The central office is also responsible for the overall supervision of the statewide structure inspection and inventory program, statewide structure load posting program, and statewide training of structure inspectors.

For a successful statewide bridge safety inspection and management program various parties at different levels are involved. These include Federal Highway Administration (FHWA), Central Office, Office of Structural Engineering, District Offices, and Bridge Owners.

Each state has strict rules for the qualification of inspection program personnel. A number of qualification requirements are set by the State Department and Federal Authorities. Qualifications are required for program managers, reviewer for safety inspection, safety inspection team leader, safety inspection team member.

Inventory and inspection records are to be prepared and maintained in accordance with NBIS. Bridge inspection reports and some inventory data are warehoused by the department's bridge management system, and electronic database. These records are also forwarded to the Federal Highway Administration (FHWA) [9].

2.2.5 Ethiopian Practice

Most of the bridges in Ethiopia are old and currently suffering from heavy traffic load, which they were not originally designed for. In general, due to aging, overloading, negligence in maintenance and different types of damages, most of the bridges in Ethiopia have problems. Furthermore inspection reports recently made on some bridges exhibit that the problem is critical and needs urgent intervention [6].

It is reported that bridges in Ethiopia have been neglected for a long time in-terms of inspection and maintenance. According to the Road Sector Development Program (RSDP), which was launched in 1997 to speed up improvement and expansion of Ethiopian road network, only 21% of nearly 25000 km road network is found in 'Good' condition [6].

In order to overcome this problem, in August 1999, in support of RSDP, the Japanese Government -through JICA has dispatched an Expert who was mainly assigned to render Technical Assistance in preparation of bridge inspection guideline which was completed in 2001. After preparation of the Inspection Guideline ERA's Bridge Engineers were highly motivated to develop a computer program for management of bridges in Ethiopia. The purpose of the software is to enable the concerned Authorities to be systematically informed on the condition of bridges and their maintenance needs in the short and long run, besides producing all other bridge relevant information, followed by the Nation wide bridge Inventory and Inspection project that was the first of its kind in the country.

After developing Bridge Management System (BMS) software, National Bridge Inventory and Inspection project was launched and it took place in 18 months of duration to cover all roads under the Federal and Regional Road Authorities in two phases [6].The first phase was from June 2005-April 2006 for Federal roads and the second phase was from April 2008_August 2008 for regional roads. This project was undertaken by 5 different local private firms and the project cost, 1.41 Million USD, was financed by the Ethiopian Road Fund Office [6].

As a result for the first time since the establishment of Ethiopian Roads Authority in 1943, the amount of the bridge stock and their condition came to be known. It is reported that currently almost all technical and administration data of all bridges in Ethiopia including photographs have been entered into the BMS software.

Table 2.1 – Roads and Segments Data in Ethiopia [6]

It. No	Road Agency	Road Length, km*	Bridges, Length≥4.0 m	Culverts, Length<4.0 m
1	ERA – Federal Roads	20429	2955	25457
2	Tigray RA	1197	248	2629
3	Afar RRA	455	35	493
4	Amhara RRA	2197	219	2231
5	Oromia RRA	5209	440	5014
6	Somali RRA	456	6	Not completed to date
7	Southern RA	2947	251	1938
8	Gambella RRA	436	15	422
9	Benishangul Gumz RRA	596	66	512
10	Harere RA	150	7	175
11	Diredawa RA	200	15	200
12	Addis Ababa City RA**	3896	150	1500
Total		36087	4407	40567

*The Road Length shown consisting of asphalt and gravel, does not cover under constructions and community roads.
**Estimated

It is also noted that in order to assure sustainability of the recently started BMS implementation, ERA will contract out soon the Bridge Management Support Service Project to be handled by local private firms. The project life span is envisaged to be 3 years since January 2009 [6].This will be accomplished by conducting regular bridge inspection, updating bridge database, making improvement assessment, preparation of bridge and culvert improvement master plan, revising bridge repair specification as well as supervising the improvement work.

In order to facilitate the proposed work, ERA has already distributed to all District Offices Bridge inspection manuals (in English and Amharic languages), Bridge and Culvert data collection standard formats, Bridge Management System and Culvert Management System software.

2.3 Ratings and Load Posting

2.3.1 General

Each bridge carrying vehicular traffic shall be rated to determine its safe load carrying capacity. If it is determined that the maximum legal load configurations exceeds the load allowed at the bridge, then the structure shall be posted for load restriction. The notice shall caution all persons against driving on the bridge a loaded conveyance of greater weight than the bridge's carrying capacity. The use of warning signs shall be based on an engineering study.

Bridges that do not have sufficient capacity under the legal loads rating should be posted for load restriction for immediate solution and considered for strengthening for the future.

Girder supported concrete deck slabs or metal decks that are carrying normal traffic satisfactorily need not be routinely evaluated for load capacity. Timber decks that exhibit excessive deflection under normal traffic loads are candidates for further evaluation and often control the rating. Members of substructures need not be routinely checked for load capacity. Substructure elements such as pier caps and columns should be checked in situation where the Engineer has reason to believe that their capacity may govern the load capacity of the entire bridge. Typically, only the main live load-carrying superstructure members are analyzed and rated to determine the need for a bridge restriction.

The Engineer must determine if the condition and/or structural makeup of bridge elements other than the main live load carrying members control the bridge's capacity safety to carry live loads.

There are three different types of bridge ratings. These are:

- Condition Ratings
- Appraisal Ratings
- Load Ratings

2.3.2 Condition Ratings

Condition ratings are the status of the various components of a bridge as noted by the field inspector. They are objective and not opinions.

Condition ratings based on field inspections are snapshots in time and cannot be used to predict future conditions or behavior of the structure. However, condition ratings based on inspections along with written comments by a field inspector act as the major source of information on the status of a bridge. Condition ratings also help planning for necessary repairs or modifications. In addition, the condition ratings are important when performing over-weight permit evaluations.

Condition ratings reflect deterioration or damage and do measure design deficiency. For instance, an old bridge designed to a low load capacity but with little or no deterioration may have excellent condition ratings while a newer bridge designed to modern loads but with considerable damages will have lower condition ratings.

Each element of a bridge is rated based on independent consideration. For instance, poor or deficient secondary members (bracing, diaphragms, etc.) in a superstructure may cause the superstructure component to have a poor rating even though the main members show no significant deterioration. The summary component rating must be the least of the element ratings comprising that component.

Condition ratings are still a matter of judgment, which should be made based on experience, knowledge, and consistency with other structures with the same deterioration.

2.3.3 Appraisal Ratings

Appraisal ratings consider the field condition, waterway adequacy, geometric and safety configurations, structural evaluation, and safe load capacity of the bridge. The intent for appraisal ratings is to compare the bridge to a new structure built to current standards.

Six features are evaluated for their effect on the safety and serviceability of the bridge and its approaches. These are:

- Traffic Safety Features
- Structural Evaluation
- Deck Geometry
- Under clearances
- Waterway Adequacy
- Approach Roadway Alignment

2.3.4 Load Ratings

The load rating is a measure of bridge live load capacity and it has two commonly used categories:

- Legal load rating: it is that load, including loads in multiple lanes that can safely utilize the bridge for an indefinite period of time. This rating is conducted to make sure the adequacy of the bridge to current traffic loads.

- Permit: it is the maximum permissible live load that can be placed on the bridge. This load rating also includes the same load in multiple lanes. This rating is necessary when special vehicles or heavy loads need to cross the bridge.

Load ratings are computed using the AASHTO Legal Loads (TYPE 3, TYPE 3-2, TYPE 3-3, and Legal Lane Load). These loading patterns, together with the prescribed live load factors, give a realistic estimate of the maximum live load effects of a variety of heavy trucks in actual traffic. A detailed description of these loads is discussed in chapter 3.

2.3.5 International Practice

In America the board of county commissioners together with the county engineer or an engineer to be selected by the board, or the director of transportation, may ascertain the safe carrying capacity of the bridges on roads or highways under their jurisdiction. Additional information for warning signs in each state is provided in the Manual of Uniform Traffic Control Devices. For example in Ohio state, such additional information is provided in the Ohio Manual of Uniform Traffic Control Devices (OMUTCD) presenting Ohio Department of Transportation policies, standards, guidelines, practices and procedures concerning the design, construction, operations and maintenance of various types of traffic control signing [9].

Each bridge is load rated on every inspection cycle. The load rating is done based on AASHTO Legal Loads. When the bridge is found to be deficient either posting or strengthening activity is carried out.

Implementation of Posting

A. When the Load Rating of the bridge is determined to be less than 100% of legal load and the bridge cannot be strengthened immediately to a rating of 100% or above, the following procedures are used:

1. The program manager will:

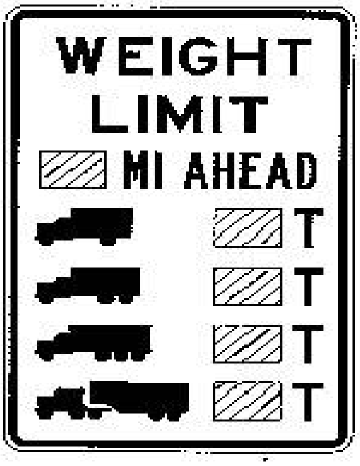
a. Establish a rating and submit to the Structure Rating Engineer, a written request for the bridge posting.
b. After the Director signs the posting request, the District will prepare, erect and maintain all necessary signs until the bridge is either strengthened or replaced.

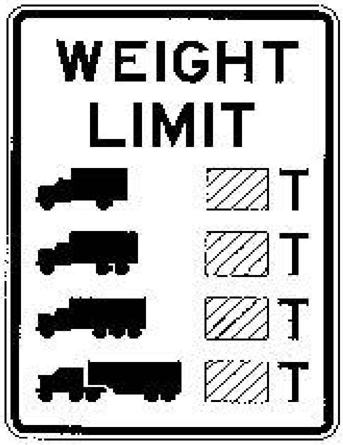
2. The program manager shall update all Bridge Inventory and Inspection records to show the latest official posted capacity.
3. After the posting request is signed, the Structure Rating Engineer will send a copy to the: program manager; Manager of Hauling Permits Section of the Office of Highway Management; Superintendent of State Highway Patrol; Executive Director Trucking Association; the Board of County Commissioners; and the County Engineer where the structure is located.

B. Special treatment shall be applied to legal load ratings of 95% or higher and also to legal load ratings of 15% or less as follows:

1. Because of the use of some judgmental data in the rating computations, bridges with a calculated load reduction of 5% or less, after rounding, shall not be posted. These structures shall be rated at 100% of legal load.
2. For calculated load reductions of 85% or more, after rounding, the bridge must be considered for “closing” to all traffic until it can be rehabilitated or replaced.

C. Where posting of a bridge is determined necessary and no unusual or special circumstance at the bridge dictates otherwise, Standard regulatory signs shall be placed in sufficient numbers and at the specific locations required below.





Bridge Ahead Sign   Bridge Weight Limit Sign

Figure 2.1 Example of Standard Wording to be used on Signs.

- Bridge Ahead signs shall be erected at intersecting state roads located just prior to the bridge to allow approaching vehicles to by-pass the bridge or turn around safely with a minimum of interference to other traffic.
- Bridge Weight Limit signs shall be erected at each end of the structure.

2.3.6 Ethiopian Practice

2.3.6.1 General

ERA has stated that maintaining a full stock of bridges in good condition to secure the safety of traffic at anytime is its utmost important duty [6].The Bridge Management System software can be used to access the bridge condition for prioritization purpose which is vital for improvement intervention. Ethiopian bridges are categorized as in "GOOD", "FAIR", and 'BAD' condition as show in Table 2.2.

Table 2.2 – Condition Evaluation of Bridges [6]

Condition of Bridges	Federal Roads – ERA Sep, 2006 (%)	Anticipated Improvement plan, 2010 (%)	Anticipated condition of bridges in absence of intervention, 2010 (%)
GOOD Still sound, adequate and functional	54.00	79.00	41.00
FAIR Inadequate and require rehabilitation	36.00	20.00	36.00
BAD Critical, requiring immediate intervention	10.00	1.00	23.00

ERA has proposed that all bridges under severe damage condition are selected and packaged for improvement action. The improvement action can vary from minor repair to total replacement [6].

2.3.6.2 Damage Rating

In order to judge the necessity of repair work, observations of damage or deterioration are categorized into three ranks as follow:

Rating 'A'

This rating implies that urgent repairs are necessary to secure the safety of traffic and to avoid major damage or deterioration of the structure.

Urgent repair or restoration of a damaged bridge component is required to avoid hazard to traffic or third parties as follows:

- Serious damage or hindrance to traffic which causes a reduction in available traffic lanes, leading to traffic disruption on the roads.
- Serious damage which affects the safety of third parties or the possibility that the damage may increase in the near future again affecting the safety of third parties.
- Serious damage which affects the loading capacity of the structures or structural components.

Rating 'B'

In this rating, repairs are necessary but not urgent. Repairs are required due to the existence of serious damage or deterioration, which affect the function, or durability of the structure.

Rating 'C'

The damage, defect or deterioration is minor and the remedial measures necessary are associated with routine maintenance to ensure structural integrity and provide a good level service to daily traffic.

In order to show this rating level the following table illustrates the rating level for cracking of deck slab.

Table 2.3 Sample of Damage Rating [6]

Damage Item	Rank 'A'	Rank 'B'	Rank 'C'
Cracking	Single crack width of more than 5 mm or multiple cracks of width more than 3 mm, together with water leakage, free lime or salt. In bearing area, single crack of more than 3 mm and multiple crack width of 1mm which causes reduction in loading capacity.	Single crack width of more than 3 mm or multiple cracks of width more than 1 mm, together with slight water leakage. In bearing area, single crack width of more than 1mm and multiple crack of width less than 1 mm, which causes reduction in durability.	Single crack width of more than 1 mm or multiple crack of width less than 1 mm in bearing area single crack width of less than 1 mm with no water leakage.

2.3.7 Recommended Practice

So far only condition rating of bridges is done on Ethiopian bridges. This is to say that in order to determine the need for bridge maintenance or replacement, damage inspection is conducted and rating is given based on the current condition of the bridge without assessing its load carrying capacity. But in order to get the full picture of the condition of a bridge, load rating along with condition rating is important.

Load rating is conducted in order to check the bridge capacity to carry current traffic loads. A detailed description of this rating is given in ERA's Bridge Design Manual. This rating is done by loads that are said to represent the current traffic condition of Ethiopia.

Load rating should be incorporated with the bridge inventory and inspection records. The primary aim of this thesis is to address this shortcoming. A computer program has been developed that will enable to load rate simple span slab and girder bridges based on their current condition. A detailed description of this program is given in the fourth chapter.

Since bridge inspection practice in Ethiopia is only of recent vintage much work is required in order to assure the sustainability of the program and to achieve the desired goal. As can be learnt from other countries, the concerned Authorities needs to make sure that bridge inspection is conducted on regular basis, inspection procedure are strictly followed, inspections are conducted by qualified personnel, bridge inventory and inspection data are well documented and updated, and damaged bridges are maintained or replaced as necessary and make sure that load restrictions are observed.

Procedure for Rating

A. Load ratings for each bridge is determined in the following manner:

1. A careful field inspection of the bridge is made by the program manager and/or other qualified structural engineer to determine its condition, and the percent of effectiveness of the various members for carrying load. All information shown in the Bridge Inventory and Inspection Records shall also be carefully checked and revised as necessary to show the current condition of the bridge.

   Using pertinent current information, the program manager and/or other qualified structural engineer shall determine the Legal Load ratings for the structure. The Legal Load shall be determined by Load and Resistance Factor Methods and shall be expressed in terms of the Percent of Legal Loads.

2. The yield stresses for the construction materials in older bridges, for which plan information is not available, can generally be conservatively estimated using the data based on date of construction.
3. The program manager shall submit to the Structure Rating Engineer a complete condition report and the original copy of the rating calculation sheets or computer input data sheets for each bridge under his/her jurisdiction.

4. The Structure Rating Engineer shall review the submitted material and return a copy of the final calculations or computer output to the Program Manager, along with any recommendations concerning the proposed ratings.

3. Strength Evaluation of Existing Bridges

3.1 General

In this chapter an established methodology for rating existing bridges will be discussed. This methodology is based on the Load and Resistance Factor Design concept. Conservative assumptions are made in each step of a strength checking procedure to safeguard against failure when the expected worst possible conditions occur during the lifetime of a structure. The rating methodology has been developed to maintain consistent safety levels for any uncertainties. Options for incorporating site specific traffic and loading data and higher levels of effort by the engineer are introduced since these lead to a reduction in the overall uncertainty [5]. The methodology can be used to evaluate almost all existing bridge types in Ethiopia.

The procedure for rating existing bridges requires knowledge of the physical conditions of the bridge and the applied loading. Detailed investigation of the structure's physical condition, continuing attempt to alleviate any sign of deterioration, knowledge of traffic conditions including signs of overweight vehicle combinations combined with accurate structural analysis gives confidence on the rating of a bridge. The evaluation of a structure is based on the simple principle that the available capacity of a structure to carry loads must exceed the required capacity to support the applied loadings.

The rating check is done by comparing the factored load effects (both dead and live) with the factored resistance at all critical sections. The output is a rating factor, which determines the suitability of the given bridge for the loads under considerations.

To mention some of the advantages of this methodology:

1. It provides uniformly consistent procedures for evaluating existing bridges.
2. It permits suitable flexibility in making evaluations.
3. It provides uniform levels of reliability developed from performance histories.
4. It is based on extensive truck traffic and bridge response data.
5. It permits introduction of site specific data into the evaluation in a rational and consistent format.
6. It permits different levels of effort that involve progressively more work; with correspondingly greater rewards in terms of more beneficial ratings.
7. It includes the same nominal dead and live load effect calculations and resistances as in the design of new bridges.
8. It allows distinction between evaluation of redundant and nonredundant components.

The methodology is intended to produce rating factors for routine evaluation and posting considerations.

3.2 Loading Patterns Classification and Ratings

3.2.1 The Rating Equation

The evaluation is carried out with a comparison of the factored live load effects and the factored strength or resistance. The live load factor accounts for uncertainties in expected maximum vehicle loading effect, impact and distribution of loads during a time period between inspections. The resistance factor accounts for uncertainties in strength prediction theories, material properties and deterioration influences over time periods between inspections.

The rating procedure is carried out for all strength checks (moment, shear, reaction, etc.) at all potentially critical sections with the lowest value determining the rating factor for the entire span.

The rating equation is:

$$\emptyset R_n = \gamma_D D + \gamma_L (RF) L (1 + I) \qquad (3.1)$$

or:

$$RF = \frac{\emptyset R_n - \gamma_D * D}{\gamma_L * L(1 + I)} \qquad (3.2)$$

Where:	RF	=	rating factor (the portion of the rating Legal Truck allowed on the bridge)
	ϕ	=	resistance factor
	R_n	=	nominal resistance
	γ_D	=	dead load factor
	D	=	nominal dead load effect
	γ_L	=	live load factor
	l	=	nominal traffic live load effects
	L	=	nominal live load effect
	I	=	live load impact factor

The rating factor is the ratio of the safe level of loading to the load produced by the nominal or standard vehicle. It shall be used in the consideration of posting levels and/or the consideration and justifications for future repairs or replacement. In determining load and resistance factors for the rating equation, the following steps shall be carried out in evaluating a bridge span:

1) Collection of information
2) Selection of nominal loadings and resistances
3) Distribution of loads
4) Selection of load and resistance factors
5) Calculation of rating factors

The evaluator/designer should note that potential improvement in the rating factor may come from selecting options in each step. These generally provide a less conservative factor provided additional evaluation effort is performed and no unsatisfactory information is uncovered.

The basic structural engineering equation states that the resistance of a structure must equal or exceed the demand placed on it by loads. Stated mathematically:

$$R \geq \Sigma Qk \tag{3.3}$$

Where: R = resistance

Q_k = effect of load k

The solution of this simple equation encompasses the whole art and science of structural engineering including the disciplines of strength of materials, structural analysis and load determination. This equation applies to design as well as evaluation. In structural evaluation, the objective is to determine the maximum allowable live load. In the case of bridge evaluation, this usually means the maximum vehicle weight.

Any rational and tractable approach to the analytical solution of the basic structural engineering equation requires that the modes of failure be identified to establish the resistance. The location, types and extent of the critical failure modes must be determined. The checking equation must be solved for each of these potential failure modes.

Since neither resistance nor the load effect can be established with certainty, safety factors must be introduced that give adequate assurance that the limit states are not exceeded. This shall be done by stating the equation in a load and resistance factor (LRFD) format.

The basic rating equation used in the guidelines is simply a special form of the basic structural engineering equation with load and resistance factors introduced to account for uncertainties that apply to the bridge evaluation problem. It is written as follows:

$$RF = \frac{\emptyset R_n - \sum_{i=1}^{m} \gamma_i^D * D_i - \sum_{j=1}^{n} \gamma_j^L L_j (1 + I)}{\gamma^{LR} L_R (1 + I)} \tag{3.4}$$

Where:

Where:	RF	=	rating factor (the portion of the rating Legal Truck allowed on the bridge)
	ϕ	=	resistance factor
	m	=	number of elements included in the dead load
	R_n	=	nominal resistance
	n	=	number of live loads other than the rating vehicle
	γ_i^D	=	dead load factor for element "i"
	D_i	=	nominal dead load effect of element "i"
	γ_j^L	=	live load factor for live load "j" other than the rating vehicle(s)
	L_j	=	nominal traffic live load effects for load "j" other than the rating vehicle(s)
	γ^{LR}	=	live load factor for rating Legal Truck
	L_R	=	nominal live load effect for the rating Legal Truck
	I	=	live load impact factor

The maximum permitted traffic live load effect will be the total resistance minus the effect of loadings other than the rating Legal Truck. This will include dead loads, non-vehicular live loads, and, in the case of unsupervised permit loading, the vehicular live load and the impact of normal traffic that could mix with the rating Legal Truck.

3.2.2 Collection of Information

Before the load rating of a specific bridge can be conducted, a certain amount of information has to be gathered. The extent to which the engineer is required to collect information will have a direct influence on the load rating of the bridge due to the selection of the proper category for the load and resistance factors.

The following items should be noted since they can have an influence on the selection of load and resistance factors.

1. Deck condition – Field tests have shown that the single most important factor affecting impact is roadway roughness and any bumps, sags, or other discontinuities which may initiate or amplify dynamic response to truck passages. Any of these surface factors should be noted during a bridge inspection.

2. Structural Condition - Signs of recent deterioration in structural members, which may go unchecked and increase the likelihood of further section capacity loss before the next cycle of inspection and rating should be noted. Conversely, maintenance efforts to mitigate such deterioration should also be noted. An allowance for structural deterioration should note whether this is either an expected or conservative estimation since further deterioration may increase the uncertainty regarding reliable section properties and strength during the next inspection interval.

3. Traffic Condition - The expected loading during the inspection interval is affected by the truck traffic at the site. In the best instance, data will be available from traffic surveys including objective truck weight operations. Alternatively, advice should be sought from the traffic division regarding truck traffic volume, composition, permit activities, overload sources, and degree of enforcement.

3.2.3 Selection of Nominal Loading and Resistances

Loads consist of concentrated or distributed forces that are applied directly to the bridge or result from deformations or the constraint of deformations. For bridge evaluations, the most important loads are dead load and vehicular live load plus its accompanying dynamic effects, since each of these loadings induce high superstructure stresses. Loadings other than dead load and traffic live load usually do not result in significant bending or shear in the superstructure. Since the critical mode of failure for traffic live load almost always occurs in the superstructure, other types of loads will seldom affect the live load capacity of the bridge. When other combinations of loads can affect the capacity of the bridge such as when substructure components can fail due to traffic live loading, proper investigation should be carried out to check its safety.

3.2.4 Dead Loads

The dead load shall be estimated from data available from the inspection at the time of analysis. The dead load factor accounts for normal variations of material densities and dimensions. Nominal dimensions and densities shall be used for calculating dead load effects. For overlays, either cores shall be used to establish the true thickness or an additional allowance of 20% should be placed on the nominal overlay thickness indicated at the time of analysis. The recommended unit weights of materials used in computing the dead load shall be as provided in Table 3-1 [5].

Table 3-1 Unit Weights of Materials

MATERIAL	UNIT WEIGHT [kN/m^3]
Asphalt surfacing	22.5
Concrete, plain or reinforced (normal weight)	24.0
Steel	79.0
Cast iron	72.0
Timber (treated or untreated)	8.0
Earth (compacted), sand gravel or ballast	18.0

The dead load of the structure is computed in accordance with the conditions existing at the time of the analysis.

Dead load can usually be determined more accurately than any other type of loading. One major source of error is failure to consider some of the elements that will contribute to dead load. Some items that are often overlooked are:

- Wearing surfaces
- Railings and Utilities
- Structure modifications not shown on plans

Other items that can affect the calculation of dead load are dimensional variations in the concrete section and variations in the unit weight of material. This is because the material unit weight may not be constant along the element and/or its dimension may vary because of construction defects.

The prescribed dead load factor recognizes the uncertainties in the nominal dimensions and analysis of dead load effects. Overlay thicknesses are a source of greater uncertainty in the dead load so they are assigned a 20% higher load factor unless cores or more detailed measurements are made.

3.2.5 Live Loads

The guidelines specify the number of vehicles to be considered on the bridge at any one time. These numbers are based on an estimate of the maximum likely number of vehicles under typical traffic situations. When unusual conditions exist, adjustments to the specified number of vehicles should be made.

Highway vehicles come in a wide variety of sizes and configurations. No single vehicle or load model can accurately reflect the effects of all of these vehicles. The variation will usually be greater than the variation in dead load effect. To minimize this difference, it is necessary to select a rating Legal Truck with axle spacing and relative axle weights similar to actual vehicles. Three Legal Trucks shown in Figure 3-1 to 3-3 are recommended as evaluation vehicles. These vehicles, together with the prescribed live load factors, give a realistic estimate of the maximum live load effects of a variety of heavy trucks in actual traffic.

The moving loads to be applied on the deck for calculating maximum nominal live loading effects shall be the three Legal Trucks. The spacing and axle weights chosen for these vehicle types were selected from actual trucks. It is believed that these typical vehicles correspond better to existing traffic and will provide more uniform reliability than the old standard AASHTO H or HS design trucks. Hence, the latter are not recommended for bridge posting purposes.

In computing load effects, one Legal Truck shall be considered present in each lane. The positioning of the vehicle in each lane shall be according to the Requirements for new bridge design. It is unnecessary to place more than one vehicle in a lane since the load factors shown below have been modeled for this possibility. These load factors shall be considered applicable for spans up to 60m.

For longer spans, a lane loading is specified in the evaluation, shown in Figure 3-4. Reduction factors for live loading of more than two traffic lanes are provided. These rationally account for the lower possibility of such occurrences.

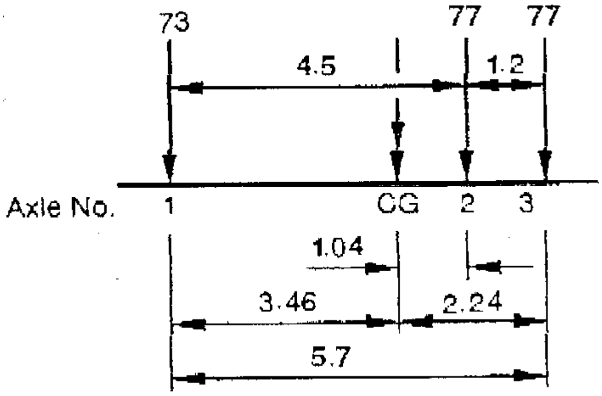


Figure 3-1 Truck Type 3-1 Unit Weight = 227 kN

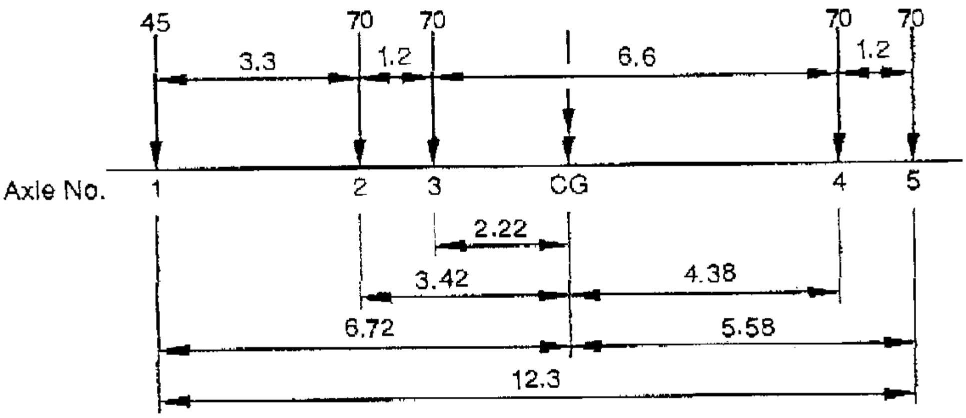


Figure 3-2 Truck Type 3-2 Unit Weight = 325 kN

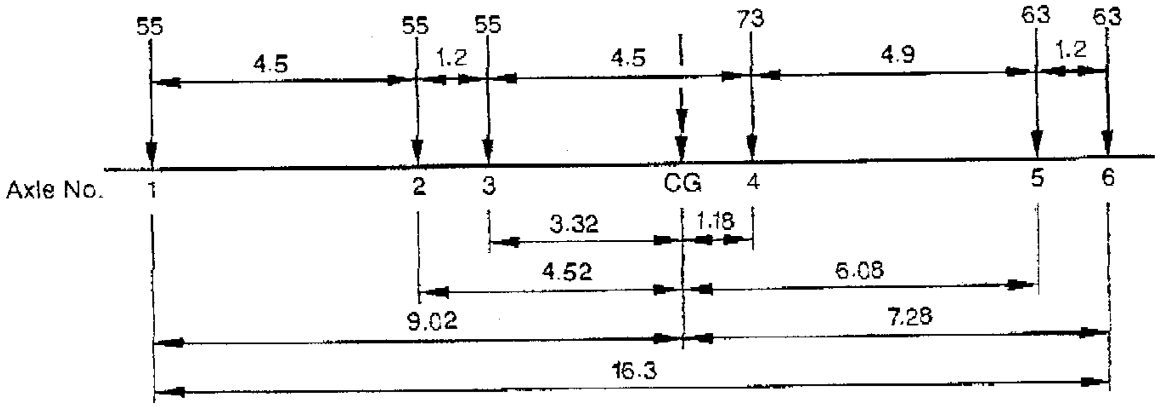


Figure 3-3 Truck Type 3-3 Unit Weight = 364 kN

Figure 3-4 The Legal Lane Loading (Mainly for Large Spans)

For longer spans, the Legal Lane Loading given in Figure 3-4 will govern the evaluation (up to 90 m). This is a combination of a vehicle load and a uniformly distributed load of 3 kN/m. For all span lengths where the rating factor is less than one, it shall be necessary to place more than one vehicle in each lane. In lieu of this, the evaluator should check the lane loading for all span lengths together with the rating Legal Truck as shown in Figure 3-4.

In checking special permits, the actual vehicle weights and dimensions shall be used. If the number of such permits in one year is frequent, then it shall be assumed that two lanes are occupied by such a vehicle. Otherwise, standard vehicles shall be placed in the other lanes. When the engineer determines that conditions of traffic movement and volume warrant it, the standard vehicles shall be eliminated. Upon special investigation, the load factor for a controlled permit use is reduced below the value taken for ordinary traffic conditions.

Since overload permissible vehicles typically have very different axle configurations, it is very important that this be considered when issuing permits.

The probable maximum sidewalk loadings should be used in calculations for safe load capacity ratings. This loading will vary from bridge to bridge, depending generally upon its location. Because of this variation, the Engineer must use his judgment to make the final determination of the unit loadings to be used.

3.2.6 Impact

An impact allowance shall be added to the static loads used for rating. Impact values are used for design of new bridges to reflect conservative conditions that may possibly prevail under certain circumstances. Under an enforced speed restriction, impacts shall be reduced.

Impact loads are taken to be primarily due to the roughness or unevenness of the road surface, especially the approach spans. Three values of impact factors are provided by correlating the roughness of the surface to the deck conditions survey values. This information is more likely known during evaluation than in the original design.

For smooth approach and deck conditions, the impact shall be taken as 0.10. For a rough surface with bumps, a value of 0.20 should be used. Under extreme adverse conditions of high speed, spans less than 12m and highly distressed pavement and approach conditions, a value of 0.30 should be taken [5].

If such a judgment cannot be made, refer to the bridge inspection report and relate impact to the condition of the wearing surface.

Table 3-2 Condition of Wearing Surface and Impact Value [5]

WEARING SURFACE	DESCRIPTION	IMPACT
1 - Good condition	No repair required	0.1
2 - Fair condition	Minor deficiency, item still functioning as designed	0.1
3 - Poor condition	Major deficiency, item in need of repair to continue functioning as designed	0.2
4 - Critical condition	Item no longer functioning as designed	0.3

3.2.7 Resistances

Nominal strength calculations shall take into consideration the observable effects of deterioration, such as loss of concrete or steel cross-sectional area, loss of composite action or corrosion.

Concrete: The strength of sound concrete shall be assumed to be equal to either the values taken from the plans and specifications or the average of construction test values. When these values are not available, the ultimate stress of sound concrete shall be assumed to be 25 MPa. A reduced ultimate strength shall be assumed (no less than 15 MPa, however) for unsound or deteriorated concrete unless evidence to the contrary is gained by field-testing [5].

Reinforcing Steel: The area of tension steel to be used in computing the ultimate flexural strength of reinforced concrete members shall not exceed that available in the section or 75 percent of the steel reinforcement required for a balanced condition. The steel yield stresses to be used for various types of reinforcing steel are given below.

Table 3-3 Reinforcing Steel Yield Stresses [5]

Reinforcing Steel	**Yield Stress F_y (MPa)**
Unknown steel (prior to 1954)	228
Structural Grade	248
Intermediate Grade 300 and unknown after 1954 (former Grade 40)	276
Hard Grade (former Grade 50)	314
Grade 420 (former Grade 60)	414
Grade 520 (former Grade 75)	517

The determination of structural resistance is one of the primary tasks in the evaluation process. In a load and resistance design (LRFD - also known as limit state) approach it is necessary to define the condition at which resistance will be determined. These should provide for similar structural performance regardless of the material or structure type.

These limit states should have a very low probability of occurrence because they can lead to loss of life as well as to major financial losses. They include:

- Loss of equilibrium of all - or part of - the structure considered as a rigid body (e.g., overturning, sliding, uplift, etc.);
- Loss of load-bearing capacity of members due to insufficient material strength, buckling, fatigue, fire, corrosion, or deterioration;

- Overall instability of the structure (e.g., P-delta effect, wind flutter, seismic motions, etc.);
- Very large deformation (e.g., transformation into a mechanism).

Determination of the true safety limit state involves very complicated and difficult analytical procedures. In most cases, the use of these procedures for routine evaluation of bridges is not economically feasible. The ultimate member capacity shall be a lower bound of the ultimate capacity in shear or in flexure. Different methods for considering the observable effects of deterioration were studied in developing the guidelines. The most reliable method available still appears to be a reduction in the nominal resistance based on measured or estimated losses in cross-sectional area and/or material strengths. An alternate approach is to calculate resistance based on plan dimensions and use a smaller capacity reduction factor.

Nominal resistances for members in the proposed guidelines are based on the load and resistance factor requirements. This resistance depends on both the current dimensions of the section and the nominal material strength. Specifications for both these factors have been provided. Options exist for incorporating data on structural conditions obtained from the site. Careful estimation of losses and deterioration are awarded a higher resistance factor. Similar gains are also given for vigorous maintenance and inspection schedules, which may prevent further deterioration during a normal inspection interval. Options also exist for obtaining more precise material strength through tests.

Structural Steel

Nominal unit stresses must depend on the type of steel used in the structural member. When tests are performed to assess yield stress, the mean values shall be reduced by 10% to produce nominal values for strength calculations. Nominal values shall be nominal strength computed without any resistance factor applied.

3.2.8 Distribution of Loads

The fraction of vehicle load effect transferred to a single member is selected in accordance with the specification given in ERA Bridge Design Manual. These values represent a possible combination of adverse circumstances. The option exists to substitute field measured values, analytically calculated values or those determined from advanced structural analysis methods utilizing the properties of the existing span(s). Loadings shall be placed in positions causing the maximum response. Further, if such a measurement or analysis is made and the expected distribution value is obtained, then this shall be adjusted by the factors shown in Table 3-4. The latter are needed to adjust for the expected bias in distribution factors for different material types.

Table 3-4 Correction Factor for Analysis* [5].

Distribution of Loads		Correction Factor*		
		Steel	Prestressed	Concrete
1	AASHTO Distribution, Chapter 13	1.00	1.00	1.00
2	Tabulated analysis with simplified assumptions	1.10	1.05	0.95
3	Refined analysis: finite elements, orthotropic plate, grillage analogy	1.07	1.03	0.90
4	Field measurements	1.03	1.01	0.90
Actual girder distribution shall be multiplied by the appropriate correction factors to obtain the girder distribution for rating.				

* Correction factors are applied if average or expected values are used for R.F. from analysis or measurements. The correction factor shall be used to increase the load factor taken from Table 3-5.

Lateral distribution refers to the fraction of the live load carried by the member under consideration. Options exist for using tabulated values, more refined analysis (e.g. finite elements) and field measurements. Each of these options involves a greater level of effort and more accuracy, so adjustments to the basic live load factors are provided. These adjustments implicitly recognize that more refined analysis may in some instances remove the implicit conservativeness present in some simplified distribution formulas and are therefore treated accordingly.

3.2.9 Selection of Load and Resistance Factors

The statistics of the dead load, live load and resistances have been determined from existing data. Based on this data, the safety implicit in current designs has been determined. The load and resistance factors provided ensure that an acceptable level of safety is achieved or exceeded.

3.2.9.1 Load Factors

The load factors used for rating of bridges are those shown in Table 3-5. These load factors are intended to represent actual traffic conditions. They are based on field data obtained from a variety of locations using weight-in-motion and other data gathering methods.

Table 3-5 Load Factors [5].

Loading	**Load Factor**
Dead Load	$\gamma_D = 1.2$
Allow an additional allowance of 20% on overlay thickness if nominal thicknesses are used. No allowance is needed when measurements are made for thickness. Live Load Category	

1	Low volume roadways (ADTT less than 1000), reasonable enforcement and apparent control of overloads	γ_L = 1.30
2	Heavy volume roadways (ADTT greater than 1000), reasonable enforcement and apparent control of overloads (not common in Ethiopia)	γ_L = 1.45
3	Low volume roadways (ADTT less than 1000), significant sources of overloads without effective enforcement (common in Ethiopia)	γ_L = 1.65
4	Heavy volume roadways (ADTT greater than 1000), significant sources of overloads without effective enforcement	γ_L = 1.80

If unavailable from traffic data, estimates for ADTT shall be made from ADT as follows: urban areas, ADTT = 25% of ADT; rural areas, ADTT - 50% of ADT. In the absence of accurate data on overloads, it shall be assumed that 30% of the trucks in Ethiopia exceed the local legal gross weigh limits.

Dead load factor reflects variations in dimensions, unit weights and methods of calculating dead load effect. The variation in the dead load of different components will depend on the accuracy with which the components can be manufactured and/or measured. The higher dead load factor for asphalt recognizes the greater uncertainty in overlay thickness.

Live load factors have been provided to account for the large uncertainty of the maximum live load effects on a structure over a period of time. A large amount of field data has been modeled to estimate the maximum live load effect together with its uncertainty. Based on this data, degree of enforcement, volume and type of traffic are isolated as the major factors influencing the live load effect.

3.2.9.2 Resistance Factors

A capacity reduction factor (ϕ) is included in the basic rating equation to account for variation in the calculated resistance. It takes into consideration the dimensional variations of the structure, differences in material properties, current condition and future deterioration, and the inaccuracies in the theory for calculating resistance.

The resistance factors for members in good condition are shown in Table 3-6, section I. The influence of deterioration, inspection and maintenance are given in section II, III and IV of this table. The nominal resistance factor is to be considered case by case as shown in Figure 3-5.

The capacity reduction factor depends on the following factors:

- Redundant and non-redundant members
- Amount of deterioration
- Type of inspection and maintenance
- Maintenance schedules

This allows the evaluation to be flexible enough and also covers a large range of types and conditions of members that shall be encountered.

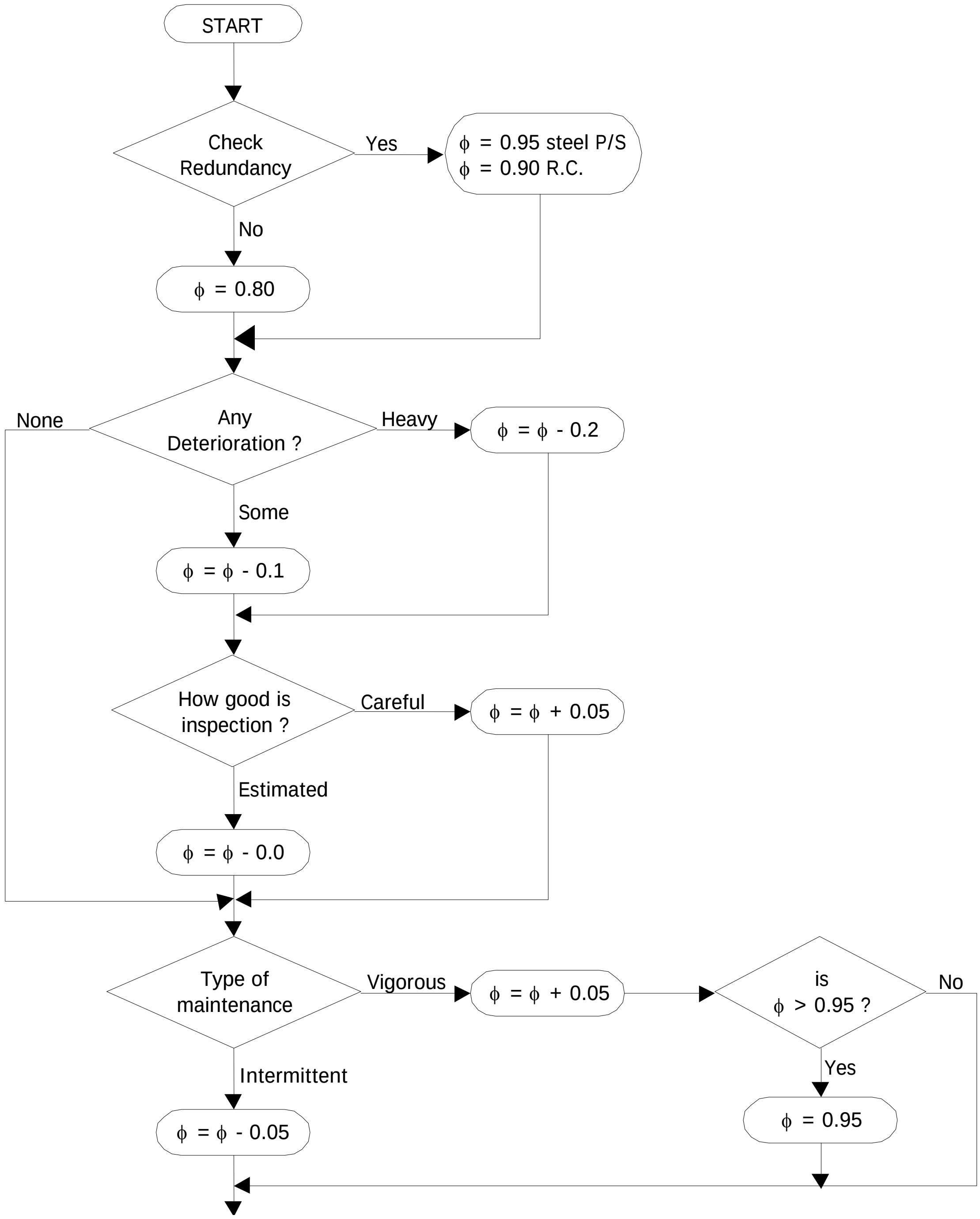


Figure 3-5 Flow Chart for Selecting Resisting Factor [5].

Table 3-6 Resistance Factors vs. Condition [5]

I. Resistance Factors - Good condition

Resistance (capacity reduction) factors are to be applied to the following for the case where members are in good condition.

- Redundant Steel Members: Φ = 0.95;
- Nonredundant Steel Members: Φ = 0.80;
- Prestressed concrete beams: Φ = 0.95;
- Reinforced concrete beams: Φ = 0.90;

These values will be modified per the conditions stated below following the flow chart shown in Figure 3.6.

II. Influence of Deterioration

1. Where field inspection and condition survey reports indicate no deterioration, the provisions of this section should not be used.

2. Where field inspection and condition survey reports indicate slight deterioration with some possible loss of section, the resistance factor values above shall be decreased by 0.1.

3. Where field inspection and condition surveys report significant deterioration and heavy section loss, the resistance factor values shall be reduced by 0.2.

4. If such information is not available then bridge records shall be used. Reduce the resistance factor values by 0.1 for superstructure condition of 5 or 6. Reduce the resistance factor values by 0.2 for a superstructure condition of 4 or less. If these reductions are made

then the next two sections should be omitted.

III. Inspection

1. Where field inspection and condition survey reports indicate no deterioration, the provisions of this section should not be used.
2. Where section losses have been carefully estimated in the calculation of remaining section areas the resistance factors shall be increased by 0.05.
3. Where material yield stress has been estimated by physical testing, a mean value of 0.90 shall be used for calculating strength together with the resistance factor contained in the design rules.

IV. Maintenance

1 Where maintenance activity is vigorous and likely to correct deficiencies which may lead to further section loss, increase Φ by 0.05.

2 Where maintenance activity is intermittent and may not correct defects that have lead to section loss, decrease Φ by 0.05.

3.2.10 Calculation of Rating Factors (RF)

The rating factor is to be calculated from Equation 3.1. If it exceeds 1.0, the span is satisfactory for the legal loads in Ethiopia. The rating factors obtained herein may also safely be applied to permit loadings. In some instances where a permit might otherwise be rejected, the live load factors contained herein shall be reduced to reflect known weight conditions associated with the permit vehicle. This reduction in load factor may depend on the degree of control of the permit and the number of permits that shall be issued.

4. Computer Program for Strength Evaluation of Bridges

4.1 General

The main aim of this thesis is to develop a computer program for strength evaluation of existing bridges in Ethiopian. This computer program has been developed using the guidelines given in ERA Bridge design manual.

Visual basic programming language was used to develop the program. The main reason for this is as follow:

- It is easy to learn and apply: it takes little time to learn the programming language. This is because the language somewhat resembles human language.
- User interface can be developed so easily. Compared to other programming languages like FORTRAN, visual basic is a powerful and easy tool to develop user interface. User interfaces can be created by just using simple windows operations like click, drag and drop etc with out writing any code.
- It is object oriented programming language. This makes the language more powerful for large programs. It is based on class and object principles. Once a class is created it can be reused by creating an instance of that class called object [12]. This makes it more manageable that those procedural programming languages like FORTRAN.

The program has been developed for Slab and Girder Concrete Bridges that are simple span. This type of bridges will cover a significant amount of bridges in Ethiopia.

The load rating is checked at critical section of members of the bridge. Shear and moments are the two forces effects checked at the critical section of the load carrying members of the bridge. Since commonly only superstructure elements may fail due to traffic loadings, load rating is conducted for these members. Hence for slab bridges load rating is checked at the slabs for they are the live load carrying members, and for girder bridges the checking is performed for the girders.

4.2 Action Effects

Three Legal Trucks and one Lane Loading are specified for rating purpose. This load are assumed to represent the current traffic condition in Ethiopia than the design load, hence the design load is not used for bridge rating. These loads are taken from actual trucks. The loads are shown in Figure 3.2, Figure 3.3, Figure 3.4 and Figure 3.5.

Load rating for a particular bridge is checked by all the legal trucks and the lane load so that the least rating is taken to be the rating of the bridge. The computer program checks the rating of a bridge for all the loading type and determines the rating value of the bridge.

4.2.1 Slab Bridges Analysis

The computer program accepts the following input parameters for analysis of slab bridges.

- **General Information**: this includes bridge identification number: bridge name: river name and material type for slab, wearing surface, railing and post.
- **Dimensions**: this include total length of superstructure: span length: total width of superstructure: clear width of superstructure: curb/edge beam width, depth: wearing surface thickness: slab thickness: railing depth, width: post height, depth, width and spacing and number of lanes.
- **Bridge Conditions**:- the following parameters are included in this category
  i. Traffic Condition: this parameter distinguishes between bridges exposed to heavy or low traffic volumes and for bridges having reasonable enforcement and apparent control of overloads and those that do not have.
  ii. Current Condition: this includes the condition of wearing surface, deck: level of inspection: maintenance activity: redundancy of the structure and cube strength of the concrete deck.

- **Reinforcement:** in this category detail of reinforcement for the middle and edge strips are provided.

After all the parameters are provided the dead load and live load effects are calculated in the following manner.

Dead Load

Dead load for each strip is calculated as follows.

1. Dead load for the middle strip is calculated from the weight of wearing surface and deck slab.
2. Dead load for the edge beam is calculate form the weight of wearing surface, deck slab, curbs/edge beams, posts and railing.

Shear force and moment effects caused by dead loads are calculated at every 5% of the span length of the bridge.

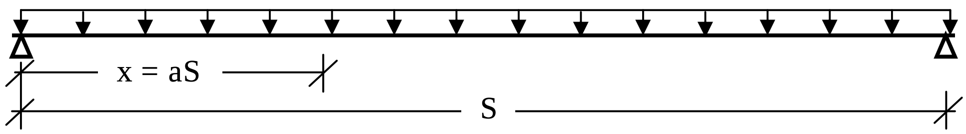


Figure 4.1 Dead Load

$$V_x = D \times S \times (1/2 - a) \quad 4.1$$

$$M_x = D \times a \times S/2 \times (1 - a) \quad 4.2$$

Where

- D = dead load
- S = span length
- a = fraction of the span length
- V_x = shear force at x = a*S, from left support

- M_x = moment at x = a*S, from left support

Table 4.1 Dead Load Shear Force and Moment along the Span of a Bridge

a	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
x=aS										
V_x										
M_x										

Live Load

The three legal loads and legal lane loading are used to determine the live load action effects. The approximate method of analysis as given in ERA Bridge design manual is used.

Lateral distribution of live load is computed as follows:

i. Middle strip

- For one lane loaded the equivalent width of longitudinal strip is:

$$E = 250 + 0.42 \times \sqrt{L_1 W_1} \qquad 4.3$$

- For more than one lane loaded, the equivalent width of longitudinal strip is:

$$E = 2100 + 0.12 \times \sqrt{L_1 W_1} \leq {}^{W}/_{N_L} \qquad 4.4$$

$$q = \frac{P}{E}\,(1 + IF) \qquad 4.5$$

ii. Edge strip

$$E_E = W_E + 300 + 0.5 \times E \quad \text{But} \quad E_E \leq \begin{cases} E \\ 1800\ \text{mm} \end{cases} \qquad 4.7$$

$$q = \frac{0.5 \times P}{E_E}(1 + IF) \qquad 4.5$$

Where: $E=E_m$ = equivalent width of middle strip (mm)

L1 = modified span length taken ≤ of the actual span or 18,000 (mm)

W1 = modified edge-to-edge width of bridge taken to be ≤ of the actual width or 18,000 mm for multilane loading, or 9,000 mm for single-lane loading (mm)

W = physical edge-to-edge width of bridge (mm)

NL = number of design lanes

P = axle load

q = live load per linear meter of equivalent width

E_E = equivalent width of edge strip

W_E = edge beam width

IF = impact factor

Each of the rating loads are applied at the bridge and the maximum of all the loads are taken. Figure 4.2 shows application of truck type 3-1 and the points (x=a*S) where the action effects are to be calculate.

Truck type 3-1

Case -1: Moment and shear at rear wheel position

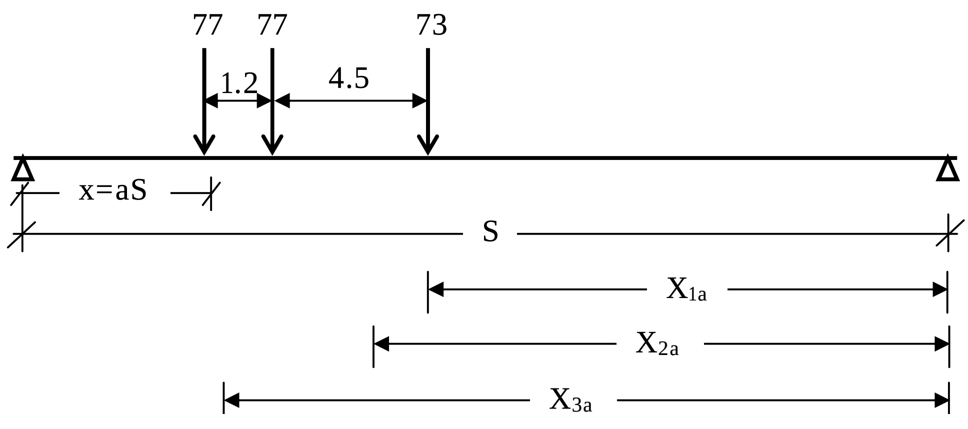


Figure 4.2 Truck Type 3 Moving on a Bridge (Rear Wheel Position)

$$V_x = (q_1 \times X_{1a} + q_2 \times X_{2a} + q_3 \times X_{3a}) \div S \quad 4.6$$

$$M_x = a \times (q_1 \times X_{1a} + q_2 \times X_{2a} + q_3 \times X_{3a}) \quad 4.7$$

a	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	(S-0.52)/S
x=aS											
V_x											
M_x											

Case -2: Moment and shear at middle wheel position

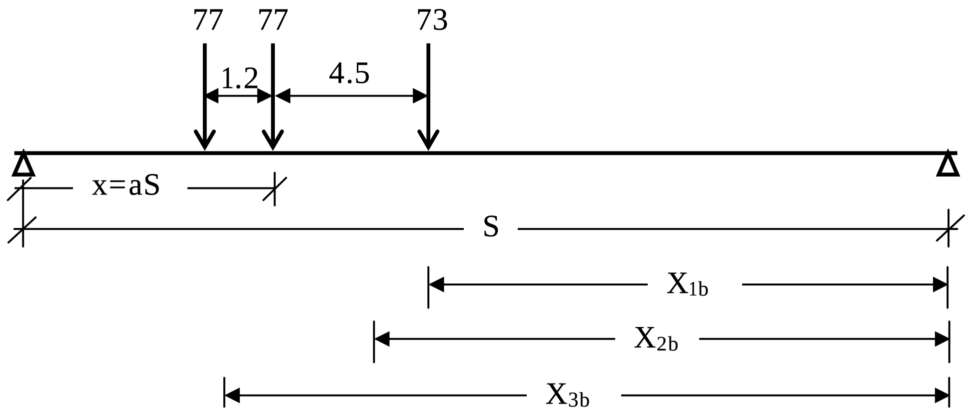


Figure 4.3 Truck Type 3 Moving on a Bridge (Middle Wheel Position)

$$V_x = (q_1 \times X_{1b} + q_2 \times X_{2b} + q_3 \times X_{3b}) \div S - q_3 \quad 4.6$$

$$M_x = a \times (q_1 \times X_{1b} + q_2 \times X_{2b} + q_3 \times X_{3b}) - 1.2 \times q_3 \quad 4.7$$

$$\text{Where: } q_1 = 73 \times (1 + IF) \div E; \quad q_2 = q_3 = 77 \times (1 + IF) \div E \quad 4.8$$

a	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	(S-0.52)/S
x=aS											
V_x											
M_x											

Similarly for the other legal truck loads and the legal lane load similar calculation is performed by using influence lines for shear and moment. In addition to the points considered in dead load calculation, the point where the absolute maximum force effects occur is taken to calculate the shear force and moment. The loads are adjusted by the appropriate live load factors which depends on different parameters discussed in chapter three.

After the dead load and live loads forces effects are calculated in this manner, the maximum force effects of all the loads both for shear and moment are taken. And the computer program does the rating calculation for all the load cases. Then it finally gives the rating of the bridge by taking the least of all the rating values.

4.2.2 Girder Bridges Analysis

In case of girder bridges the rating is done for the interior and exterior girders. The calculation is the same as that of slab bridges except that all the parameters and load distribution are those pertaining to girder bridges.

The lateral load distribution for interior and exterior girder is calculated by using the approximate method given in ERA Bridge Design Manual.

Distribution of live load per lane for moment in interior longitudinal beam

- One design lane loaded:

$$0.06 + \left(\frac{S}{4300}\right)^{0.4} \left(\frac{S}{L}\right)^{0.3} \left(\frac{K_g}{Lt_s^{\,3}}\right)^{0.1} \qquad 4.9$$

- Two or more design lanes loaded:

$$0.075 + \left(\frac{S}{4300}\right)^{0.6} \left(\frac{S}{L}\right)^{0.2} \left(\frac{K_g}{Lt_s^{\,3}}\right)^{0.1} \qquad (4.11)$$

- If number of beams is equal to three, use lesser of the values obtained from the equation above or the lever rule.

$$K_g = n\left(I + Ae_g^2\right) \quad (4.10)$$

$$n = E_B/E_D \quad (4.10)$$

Distribution of live load per lane for moment in exterior longitudinal beam

- One design lane loaded: Use lever rule
- Two or more design lanes loaded:

$$g = e \times g_{interior} \quad (4.12)$$

$$e = 0.77 + \frac{d_e}{2800} \quad (4.13)$$

- If number of beams is equal to three, use lesser of the values obtained from the equation above or the lever rule.

Distribution of live load per lane for shear in interior longitudinal beams

- One design lane loaded

$$0.36 + \frac{S}{7600} \quad (4.14)$$

- Two or more design lanes loaded

$$0.2 + 0.36 + \frac{S}{3600} - \left(\frac{S}{10700}\right)^2 \quad (4.15)$$

- If number of beams is equal to three, use the lever rule.

Distribution of live load per lane for shear in exterior longitudinal beams

- One design lane loaded: use lever rule

- Two or more design lane loaded

$$g = e \times g_{interior} \tag{4.16}$$

$$e = 0.6 + \frac{d_e}{3000} \tag{4.17}$$

- If number of beams is equal to three, use the lever rule.

Where:

L = span of beam (mm)

S = spacing of supporting components (mm)

t_s = deck slab thickness (mm)

K_g = longitudinal stiffness parameter (mm^4)

e = correction factor for distribution; eccentricity of a lane from the center of gravity of the pattern of girders (mm)

g = distribution factor

d_e = distance from the exterior web of exterior beam to the interior edge of curb or traffic barrier (mm)

n = modular ratio between beam and deck

E_B =modulus of elasticity of beam material (MPa)

E_D =modulus of elasticity of deck material (MPa)

I = moment of inertia of beam (mm^4)

e_g = distance between the centers of gravity of the basic beam and deck (mm)

A = Area of concrete (mm^2)

Lever rule

The lever rule involves summing moments about one support to find the reaction at another support by assuming that the supported component is hinged at interior supports.

When using the lever rule on a three-girder bridge, the notional model should be taken as shown in Figure 4.4. Moments should be taken about the assumed, or notional, hinge in the deck over the middle girder to find the reaction on the exterior girder.

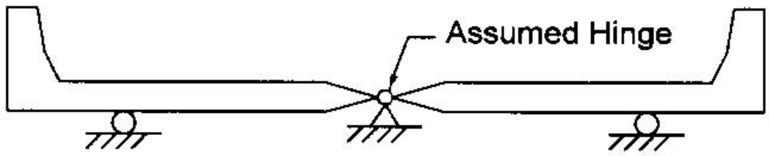


Figure 4.4 Notional Model for Applying Lever Rule to Three-Girder Bridges

4.3 Resisting Strength

Shear force and moment resisting strength are calculated based on current condition of the bridge. Standard formulas using Load and Resistance Factor Method are used to calculate these values. ERA Bridge design manual specifies that "Unless otherwise stated, the Ethiopian Building Code Standard, Vol. 2 Structural Use of Concrete, 1995, shall be used." [5].

4.3.1 Shear Strength

The shear resistance consists of a component which depends on the concrete and a component which relies on tensile stresses in the transverse reinforcement.

The nominal shear resistance is determined as the lesser of:

$$V_n = V_c + V_s \tag{4.18}$$

or

$$V_n = 0.25 f_c' b_v d_v \tag{4.19}$$

for which:

$$V_c = 0.083\beta\sqrt{f_c'}\, b_v d_v \tag{4.20}$$

and

$$V_s = \frac{A_v f_v d_v (\cot\theta + \cot\alpha)\sin\alpha}{S} \tag{4.21}$$

where: b_v = effective web width taken as the minimum web width within the depth d_v (mm)

d_v = effective shear depth (mm)

s = spacing of stirrups (mm)

β = factor indicating ability of diagonally cracked concrete to transmit tension

θ = angle of inclination of diagonal compressive stresses

α = angle of inclination of transverse reinforcement to longitudinal axis

A_v = area of shear reinforcement within a distance s (mm^2)

For non-prestressed concrete sections not subjected to axial tension and containing at least the minimum amount of transverse reinforcement specified or having an overall depth of < 400 mm, the following values shall be used:

β = 2.0

θ = 45o

With β taken as 2.0 and θ as 45^o, the expressions for shear strength become:

$$V_c = 0.166\sqrt{f_c'}\, b_v d_v \tag{4.22}$$

and

$$V_s = \frac{A_v f_v d_v (1 + \cot\propto)\sin\propto}{S} \tag{4.23}$$

4.3.2 Moment Strength

Flexural strength is calculated by taking the rectangular stress block as shown in Figure 4.5. Moreover the area of tension steel to be used in computing the ultimate flexural strength of reinforced concrete members is that available in the section or 75 percent of the steel reinforcement required for a balanced condition.

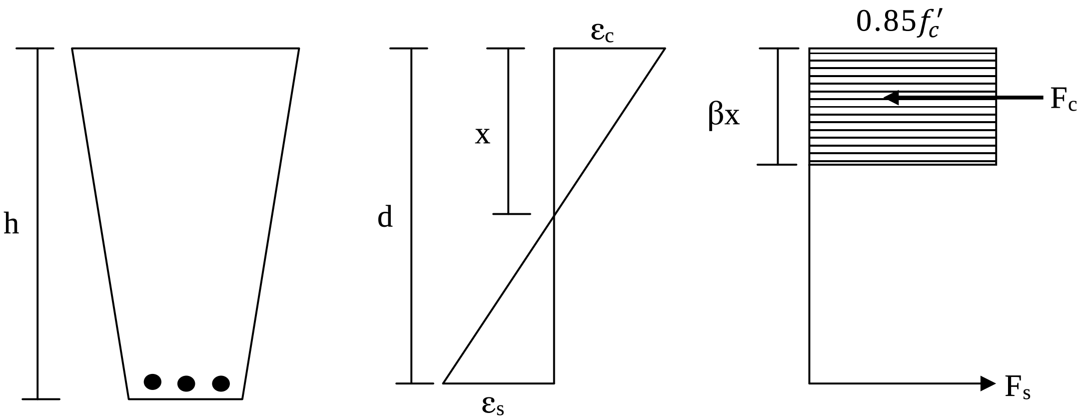


Figure 4.5 Rectangular Stress Diagram

Where

$$\beta = 0.85 \qquad if\ f_c' \leq 28\ Mpa \tag{4.24}$$

$$\beta = 0.85 - \frac{0.05}{7}(f_c' - 28) \geq 0.65 \qquad if\ f_c' > 28 \tag{4.25}$$

4.4 How the Program is Developed

4.4.1 General

As discussed before Visual Basic programming language is used to develop the software. The software is developed for slab type and girder bridges. In order to use the software the bridge has to be a simple span and the material for the supporting components should be concrete. This kind of bridge is expected to cover quite a large number of bridges in Ethiopia.

Since the software uses user interface like most windows software, data input process is easy. Once the bridge type is known, the software offers an easy way to choose the bridge type. It is also easy to switch between bridge types at any time, by just clicking a tab. If a mistake is made while data input, it can easily be corrected without restarting the software. The software also automatically detects data input errors. That is to say if the appropriate data is an integer the user can not enter floating point number.

4.4.2 User Interface

The first step in developing the software was to create the user interface. The user interface consists of tabbed windows. The tabbed windows consist of Slab Bridges and Girder Bridges as shown in Figure 4.6. These windows are further divided into a number of tabbed widows. The user can provide data into the software by the further provided tabbed windows in each category. These tabbed windows for data input consist of

- General information
- Dimensions
- Bridge Condition
- Reinforcement

Data input process is made easy and flexible. The user can give, delete and modify data at any time. This is possible because the software will do no calculation except when the run command is trigged by clicking the run button.

Finally for result viewing a results window is provided. This window gives all the input parameters and the calculated values including moment and shear strength, shear force and moment due to dead load and live load at ever 5% of the span, rating factors for every load case, and finally the rating factor of the bridge as a whole. For Girder Bridges, the same type of user interface is developed except that the data input parameters differ.

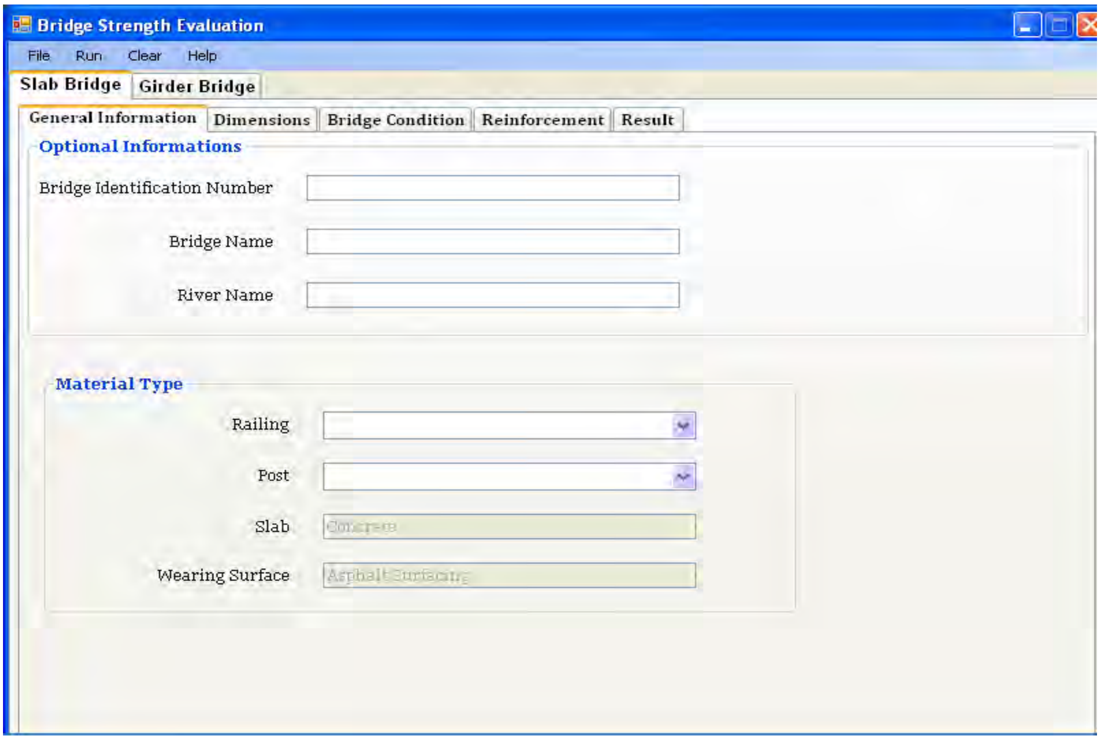


Figure 4.6 User Interface

4.4.3 The Code

Generally more than 3900 lines of code are written. The code is divided into a number of methods. The methods consist of either functions or subs. These methods perform particular tasks that form the building blocks for the entire program. The following outlines the way the code is written:

i. The first lines of code handle the data input process. These lines of code first check whether the data provided is of the correct type and if so they proceed to store the data. If the data provided is of not the correct value, that is to say for example if a user types alphabetical letters for data that should be of a number, these lines of code immediately reminds the user that the data provided is not the correct type and will delete the data. This helps to avoid run time errors.

ii. Then methods are written to determine
   a. Number of lanes
   b. Impact factors
   c. Live load factors
   d. Strength reduction factors
   e. Material unit weights
   f. Equivalent strip width
   g. Distribution factors
   h. Shear force due to dead load and live load
   i. Moment due to dead load and live load
   j. Shear force and moment strength

These methods take input parameters of bridge detailed information and current conditions to determine the relevant values. For example the method for determining strength reduction factor takes into account the input values of redundancy, bridge condition and maintenance. After all relevant information is found; it is passed to the rating equations.

iii. Then codes are written to calculate rating factors. The rating factors are determined for all the loading cases both for shear force and moment. These lines of codes then determine the least rating value to give the rating factor for the entire bridge.

iv. Codes are also written for miscellaneous operations like to display results, to save results, to clear result window, to open a new program and to exit the program. The result window shows all the input values and all the necessary calculated values.

The full source code is provided in the Appendix.

4.5 Application Using Illustrative Examples

As an example a simply supported girder bridge is load rated by the computer program. The bridge and all its detail is taken from the design example given in the appendix of ERA Bridge Design Manual. For illustrative purpose the condition of the bridge is assumed as shown in the data below. The data is given in the order the computer program accepts it.

A. Input data

1. General information

1.1. Optional information

1.1.1. Bridge identification number =

1.1.2. Bridge Name =

1.1.3. River Name =

1.2. Material type

1.2.1. Railing = **Concrete**

1.2.2. Post = **Concrete**

2. Dimensions

2.1. General dimensions

2.1.1. Total Length of superstructure =

2.1.2. Span Length (c/c of support) = **16.6 m**

2.1.3. Total width of superstructure = **8.12 m**

2.1.4. Clear width of superstructure = **7.32 m**

2.1.5. Number of lanes = **2**

2.2. Girder Dimensions

2.2.1. Girder depth = **1.3 m**

2.2.2. Girder web width = **0.38 m**

2.2.3. End of slab to center of exterior girder = **1.46 m**

2.2.4. Center to center spacing of girder = **2.6 m**

2.2.5. Number of girders = **3**

2.3. Edge beam dimensions

2.3.1. Edge beam width = **0.4 m**

2.3.2. Edge beam depth = **0.4 m**

2.3.3. Edge beam depth beneath the bottom slab level = **0.205 m**

2.4. Diaphragm Dimensions

2.4.1. Exterior diaphragm depth = **1.0 m**

2.4.2. Interior diaphragm depth = **1.0 m**

2.4.3. Width of diaphragm = **0.25 m**

2.4.4. Center to center distance between diaphragms = **8.3 m**

2.4.5. Total number of diaphragms = **3**

2.5. Slab and wearing surface dimensions

2.5.1. Top slab thickness = **0.22 m**

2.5.2. Overhang slab thickness near edge beam = **0.18 m**

2.5.3. Wearing surface thickness = **0.05 m**

2.6. Post and railing dimensions

2.6.1. Railing depth = **0.3 m**

2.6.2. Railing width = **0.25 m**

2.6.3. Post depth = **0.25 m**

2.6.4. Post width = **0.3 m**

2.6.5. Post height = **0.8 m**

2.6.6. Post spacing = **1.5 m**

2.6.7. Face of railing/curb to end of slab = **0.15 m**

3. Bridge condition

3.1. Traffic condition = **Low volume roadways (ADTT less than 1000), significant sources of overloads without effective enforcement (common in Ethiopia)**

3.2. Current condition

3.2.1. Interior girder = **Good to Fair**

3.2.2. Exterior girder = **Deteriorated**

3.2.3. Wearing surface = **Poor Condition**

3.2.4. Inspection = **Estimated**

3.2.5. Redundancy = **No**

3.2.6. Maintenance = **Intermittent**

3.2.7. Cube strength of concrete = **25 MPa**

4. Reinforcement

4.1. Interior girder reinforcement

4.1.1. Main reinforcement

4.1.1.1. Diameter = **32 mm**

4.1.1.2. Number of bars = **12**

4.1.1.3. Cover = **0.07 m**

4.1.1.4. Yield stress = **400 MPa**

4.1.2. Shear reinforcement (near support)

4.1.2.1. Diameter = **12 mm**

4.1.2.2. Spacing of bars = **130 mm**

4.1.2.3. Yield stress = **350 MPa**

4.2. Exterior girder reinforcement

4.2.1. Main reinforcement

4.2.1.1. Diameter = **32 mm**

4.2.1.2. Number of bars = **12**

4.2.1.3. Cover= **0.07 m**

4.2.1.4. Yield stress = **400 MPa**

4.2.2. Shear reinforcement

4.2.3. Diameter = **12 mm**

4.2.4. Spacing of bars = **130 mm**

4.2.5. Yield stress = **350 MPa**

After all these data are entered into the program, by just clicking the run button at the top of the window the result is displayed in the result window. The result for the above example is discussed.

B. Result

The result is displayed in the result window. The displayed result includes

- Input values. These are directly the data entered into the program. This helps for checking the data entered into the program is correct.

- Output values. These are the results calculated by the program and it includes:
    - Number of lanes
    - Unit weights
    - Dead load, live load and strength reduction factors
    - Dead load
    - Shear force and moment due to dead load
    - Live load distribution factor
    - Shear force and moment due to live load
    - Shear force and moment strength of the bridge
    - Rating factor for each load case
    - Rating factor for the bridge

For the above example the rating factors are listed below.

RATING FACTORS

Interior Girder

- Rating factor for shear
    - Truck Type 3: RF = 1.49
    - Truck Type 3_2: RF = 1.36
    - Truck Type 3_3: RF = 1.43
    - Legal lane loading: RF = 1.63
- Rating factor for moment
    - Truck Type 3: RF = 1.9
    - Truck Type 3_2: RF = 1.93
    - Truck Type 3_3: RF = 2.39
    - Legal lane loading: RF = 2.34

Exterior Girder

- Rating factor for shear
  - Truck Type 3: RF = 0.79
  - Truck Type 3_2: RF = 0.72
  - Truck Type 3_3: RF = 0.76
  - Legal lane loading: RF = 0.86
- Rating factor for moment
  - Truck Type 3: RF = 1.05
  - Truck Type 3_2: RF = 1.07
  - Truck Type 3_3: RF = 1.18
  - Legal lane loading: RF = 1.29

RATING FACTOR = 0.72

Since the condition of the exterior girder differs from the interior girder the rating value of the one differs from the other. As it can be seen the rating factor for the exterior girder is below 1.0. This is because the condition of this girder is deteriorated. This result indicates that load restriction on the bridge is necessary until the bridge is maintained to satisfy current traffic conditions and its rating factor is 1.0 or above 1.0.

5. Discussion and Conclusion

5.1 Discussion

As it can be seen from the report by ERA many bridges in Ethiopia are not in good conditions. For example on Federal Roads 36% are in fair condition requiring maintenance and 10% are in critical condition requiring immediate intervention.

Bridge inspection and maintenance activity in Ethiopia has only begun recently. It can be said that the program is in its beginning stage. Hence quite a large number of bridges are still in their bad condition.

So far little is practiced in using the actual strength evaluation procedure for rating of bridges. Even many concerned bodies conduct strength evaluation using procedures that are used to design new bridges. But the procedures for design of new bridge do not give the correct rating, mainly because the design loads does not represent actual traffic vehicles.

So far a computer program for strength evaluation of bridges is not used. This thesis has tried to solve this problem by providing a computer program. The computer program developed is simple to use. The program has an easy to use user interface that accepts input data. Results of calculation for load rating of bridges are provided by the program that can be saved for future use.

5.2 Conclusion

Since failure of many bridges is known to claim many lives and destroy public investment, proper bridge inspection and maintenance activity is important.

Many bridges in Ethiopia are found to be in bad condition. Hence, a lot of work has to be done by the responsible bodies to insure public safety and protect public investment.

Since bridge inspection and rehabilitation program started in Ethiopia recently much work is need to ensure the sustainability of the program. Concerned bodies need to provide proper training for firms and individuals that are conducting bridge inspection and maintenance activities and check their qualification. Appropriate bridge inspection time intervals should also be selected and implemented.

Strength evaluation of existing bridges is not well known in Ethiopia even by those conducting inspection and inventory of bridges. Only condition rating of bridges is practiced. But condition rating alone will not give us the full information about the bridge. Hence load rating along with condition rating should be conducted.

Strength evaluation of bridges is quite different from design of new bridges. For load rating of bridges the legal rating vehicles should be used instead of the design loads.

The computer program developed in this thesis will reduce the computational effort required for strength evaluation of bridges and it will provide a great help if incorporated with the overall bridge inventory and inspection program in Ethiopia.

.

References

1. Abnet Shoatatek. (2001). *Study on the Failure and Maintenance of Bridges in Addis Ababa – Mille Road*. Addis Ababa: Addis Ababa University.
2. American Association of State Highway and Transportation Officials. (1994). *Manual for Condition Evaluation of Bridges*. Washington, DC: AASHTO.
3. Chen, W.F., & Duan, L. (2000). *Bridge Engineering Handbook*. New York: CRC Press
4. Dellelegne Teshome. (1972). *Bridge Building Practice in Ethiopia: A Critical Appraisal of History and Modern Practice*. Addis Ababa: Addis Ababa University.
5. Ethiopian Road Authority. (2002).*Bridge design manual*. Addis Ababa.
6. Ethiopian Road Authority. (2008). *Report of Recent Bridge Inventory and Inspection Program in Ethiopia*. Addis Ababa.
7. Jensen, J., J. *History of Bridges – A Philatelic Review*. Trondheim: Norway.
8. Microsoft ® Encarta ®. (2006). *History of Bridges*. Microsoft Corporation.
9. Ohio Department of Transportation. (2006). *Manual of Bridge Inspection*.
10. Parsons Brinckerhoff. (1993). *Bridge Inspection and Rehabilitation*. New York: John Wiley & Sons, Inc.
11. Texas Department of transportation. (2006). *Bridge Inspection Manual*.
12. Thearon Willis & Bryan Newsome. (2006). *Beginning Visual Basic®2005*. Indianapolis: Wiley Publishing, Inc.

13. Wikipedia, the Free Encyclopedia. (2008). *Bridge*. Retrieved from the Web March 10, 2008. http://en.wikipedia.org/wiki/Bridge

Appendix -The Source Code

```
Imports System.io
Public Class MainForm

    'Slab Bridge Strength Evaluation - SB (Slab Bridge)
    'Variable Declaration - GeneralInformation
    Dim strTrafficConditionSB As String
    Dim strBridgeNameSB As String
    Dim strBridgeIdentificationNumberSB As String
    Dim strRiverNameSB As String

    'Variable Declaration - Material type
    Dim strRailingSB As String 'Railing material type
    Dim strPostSB As String 'Post material type

    'Varialble Declaration - Dimensions
    Dim dblLtSB As Double 'Total length of superstructure
    Dim dblSSB As Double 'Span length
    Dim dblWtSB As Double 'Total width of superstructure
    Dim dblWcSB As Double 'Clear width of superstructure
    Dim dblWetSB As Double 'Curb/edge beam width (top)
    Dim dblWebSB As Double 'Curb/edge beam width (bottom)
    Dim dblDeSB As Double 'Additional curb/edge beam depth
    Dim dblDwsSB As Double 'Wearing surface thickness
    Dim dblDsSB As Double 'Slab thickness
    Dim dblDrSB As Double 'Railing depth
    Dim dblWrSB As Double 'Railing width
    Dim dblDpSB As Double 'Post depth
    Dim dblWpSB As Double 'Post width
    Dim dblHpSB As Double 'Post height
    Dim dblSpSB As Double 'Post spacing
    Dim dblNpSB As Double 'Number of posts
    Dim intNLSB As Integer 'Number of lanes

    'Variable Declaration - Condition of Bridge Elements
    Dim strWearingSurfaceConditionSB As String 'Wearing surface condition
    Dim strEdgeStripSlabConditionSB As String 'edge strip Slab condition
    Dim strMiddleStripSlabConditionSB As String 'middle strip slab condition
    Dim strInspectionSB As String 'Inspection condition
    Dim strRedundancySB As String 'Redundancy condition
    Dim strMaintenanceSB As String 'Maintenance condition
    Dim dblCGSB As Double 'Concrete grade
    Dim dblBSB As Double 'Neutral axis depth

    'Variable Declaration - Middle strip main reinforcement
    Dim dblDbmSB As Double 'Diameter
    Dim dblSbmSB As Double 'Spacing
    Dim dblYbmSB As Double 'Yield stress

    'Variable Declaration - Edge strip main reinforcement
    Dim dblDbeSB As Double 'Diameter
    Dim dblSbeSB As Double 'Spacing
    Dim dblYbeSB As Double 'Yield stress

    'Bridge identification number - strBridgeIdentificationNumber
    Private Sub txtBridgeIdentificationNumber_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtBridgeIdentificationNumberSB.TextChanged
        strBridgeIdentificationNumberSB = txtBridgeIdentificationNumberSB.Text
    End Sub

    'Name of bridge - strBridgeName
    Private Sub txtBridgeName_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtBridgeNameSB.TextChanged
        strBridgeNameSB = txtBridgeNameSB.Text
```

```
    End Sub

    'Name of river - strRiverName
    Private Sub txtRiverName_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtRiverNameSB.TextChanged
        strRiverNameSB = txtRiverNameSB.Text
    End Sub

    'Railing material type - strRailingSB
    Private Sub cmbRailingMaterialTypeSB_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cmbRailingMaterialTypeSB.SelectedIndexChanged
        strRailingSB = cmbRailingMaterialTypeSB.Text
    End Sub

    'Post material type - strPostSB
    Private Sub cmbPostMaterialTypeSB_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmbPostMaterialTypeSB.SelectedIndexChanged
        strPostSB = cmbPostMaterialTypeSB.Text
    End Sub

    'Total length of superstructure - dblLt
    Private Sub txtTotalLengthOfSuperstructure_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTotalLengthOfSuperstructureSB.TextChanged
        'This procedure makes sure that the user provides only numbers as an imput
        Try
            dblLtSB = txtTotalLengthOfSuperstructureSB.Text
        Catch ex As Exception
            If txtTotalLengthOfSuperstructureSB.Text = String.Empty Or
txtTotalLengthOfSuperstructureSB.Text = "." Then
                dblLtSB = 0
            Else
                MessageBox.Show("Please give correct value for 'Total Length of Suprestructure'",
"Data Error")
                txtTotalLengthOfSuperstructureSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Span length - dblS
    Private Sub txtSpanLength_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtSpanLengthSB.TextChanged
        Try
            dblSSB = txtSpanLengthSB.Text
        Catch ex As Exception
            If txtSpanLengthSB.Text = String.Empty Or txtSpanLengthSB.Text = "." Then
                dblSSB = 0
            Else
                MessageBox.Show("Please give correct value for 'Span Length'", "Data Error")
                txtSpanLengthSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Total width of superstructure - dblWt
    Private Sub txtTotalWidthOfSuperstructure_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTotalWidthOfSuperstructureSB.TextChanged
        Try
            dblWtSB = txtTotalWidthOfSuperstructureSB.Text
        Catch ex As Exception
            If txtTotalWidthOfSuperstructureSB.Text = String.Empty Or
txtTotalWidthOfSuperstructureSB.Text = "." Then
                dblWtSB = 0
            Else
```

```
                MessageBox.Show("Please give correct Value for 'Total Width of Superstructure'",
"Data Error")
                txtTotalWidthOfSuperstructureSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Clear width of superstructure - dblWc
    Private Sub txtClearWidthOfSuperstructure_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtClearWidthOfSuperstructureSB.TextChanged
        Try
            dblWcSB = txtClearWidthOfSuperstructureSB.Text
        Catch ex As Exception
            If txtClearWidthOfSuperstructureSB.Text = String.Empty Or
txtClearWidthOfSuperstructureSB.Text = "." Then
                dblWcSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Clear Width of Superstructure'",
"Data Error")
                txtClearWidthOfSuperstructureSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Curb/Edge beam width (top) - dblWegt
    Private Sub txtCurbOrEdgeBeamWidthTop_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtCurbOrEdgeBeamWidthTopSB.TextChanged
        Try
            dblWetSB = txtCurbOrEdgeBeamWidthTopSB.Text
        Catch ex As Exception
            If txtCurbOrEdgeBeamWidthTopSB.Text = String.Empty Or txtCurbOrEdgeBeamWidthTopSB.Text
= "." Then
                dblWetSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Curb/Edge Beam Width (top)'",
"Data Error")
                txtCurbOrEdgeBeamWidthTopSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Curb/Edge beam width (bottom) - dblWegb
    Private Sub txtCurbOrEdgeBeamWidthBottom_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtCurbOrEdgeBeamWidthBottomSB.TextChanged
        Try
            dblWebSB = txtCurbOrEdgeBeamWidthBottomSB.Text
        Catch ex As Exception
            If txtCurbOrEdgeBeamWidthBottomSB.Text = String.Empty Or
txtCurbOrEdgeBeamWidthBottomSB.Text = "." Then
                dblWebSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Curb/Edge Beam Width (bottom)'",
"Data Error")
                txtCurbOrEdgeBeamWidthBottomSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Additional curb/edge beam depth - dblDeg
    Private Sub txtAdditionalCurbOrEdgeBeamDepth_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtAdditionalCurbOrEdgeBeamDepthSB.TextChanged
        Try
            dblDeSB = txtAdditionalCurbOrEdgeBeamDepthSB.Text
        Catch ex As Exception
```

```
            If txtAdditionalCurbOrEdgeBeamDepthSB.Text = String.Empty Or
txtAdditionalCurbOrEdgeBeamDepthSB.Text = "." Then
                dblDeSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Additional Curb/Edge Beam Depth'",
"Data Error")
                txtAdditionalCurbOrEdgeBeamDepthSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Wearing surface thickness - dblDws
    Private Sub txtWearingSurfaceThickness_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtWearingSurfaceThicknessSB.TextChanged
        Try
            If rdbNominalThicknessSB.Checked = True Then
                dblDwsSB = 1.2 * txtWearingSurfaceThicknessSB.Text
            Else
                dblDwsSB = txtWearingSurfaceThicknessSB.Text
            End If
        Catch ex As Exception
            If txtWearingSurfaceThicknessSB.Text = String.Empty Or
txtWearingSurfaceThicknessSB.Text = "." Then
                dblDwsSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Wearing Surface Thickness'", "Data
Error")
                txtWearingSurfaceThicknessSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Slab thickness - dblDs
    Private Sub txtSlabThickness_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtSlabThicknessSB.TextChanged
        Try
            dblDsSB = txtSlabThicknessSB.Text
        Catch ex As Exception
            If txtSlabThicknessSB.Text = String.Empty Or txtSlabThicknessSB.Text = "." Then
                dblDsSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Slab Thickness'", "Data Error")
                txtSlabThicknessSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Railing depth - dblDr
    Private Sub txtRailingDepth_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtRailingDepthSB.TextChanged
        Try
            dblDrSB = txtRailingDepthSB.Text
        Catch ex As Exception
            If txtRailingDepthSB.Text = String.Empty Or txtRailingDepthSB.Text = "." Then
                dblDrSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Railing Depth'", "Data Error")
                txtRailingDepthSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Railing width - dblWr
    Private Sub txtRailingWidth_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtRailingWidthSB.TextChanged
```

```
        Try
            dblWrSB = txtRailingWidthSB.Text
        Catch ex As Exception
            If txtRailingWidthSB.Text = String.Empty Or txtRailingWidthSB.Text = "." Then
                dblWrSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Railing Width'", "Data Error")
                txtRailingWidthSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Post Depth - dblDp
    Private Sub txtPostDepth_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPostDepthSB.TextChanged
        Try
            dblDpSB = txtPostDepthSB.Text
        Catch ex As Exception
            If txtPostDepthSB.Text = String.Empty Or txtPostDepthSB.Text = "." Then
                dblDpSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Post Depth'", "Data Error")
                txtPostDepthSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Post Width - dblWp
    Private Sub txtPostWidth_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPostWidthSB.TextChanged
        Try
            dblWpSB = txtPostWidthSB.Text
        Catch ex As Exception
            If txtPostWidthSB.Text = String.Empty Or txtPostWidthSB.Text = "." Then
                dblWpSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Post Width'", "Data Error")
                txtPostWidthSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Post height - dblHp
    Private Sub txtPostHeight_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPostHeightSB.TextChanged
        Try
            dblHpSB = txtPostHeightSB.Text
        Catch ex As Exception
            If txtPostHeightSB.Text = String.Empty Or txtPostHeightSB.Text = "." Then
                dblHpSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Post Height'", "Data Error")
                txtPostHeightSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Post spacing - dblSp
    Private Sub txtPostSpacing_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPostSpacingSB.TextChanged
        Try
            dblSpSB = txtPostSpacingSB.Text
        Catch ex As Exception
            If txtPostSpacingSB.Text = String.Empty Or txtPostSpacingSB.Text = "." Then
                dblSpSB = 0
```

```
            Else
                MessageBox.Show("Please give correct Value for 'Post Spacing'", "Data Error")
                txtPostSpacingSB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub lstTrafficConditionSB_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles lstTrafficConditionSB.SelectedIndexChanged
        strTrafficConditionSB = lstTrafficConditionSB.Text
    End Sub

    Private Sub cmbEdgeStripSlabConditionSB_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmbEdgeStripSlabConditionSB.SelectedIndexChanged
        strEdgeStripSlabConditionSB = cmbEdgeStripSlabConditionSB.Text
    End Sub

    Private Sub cmbMiddleStripSlabConditionSB_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmbMiddleStripSlabConditionSB.SelectedIndexChanged
        strMiddleStripSlabConditionSB = cmbMiddleStripSlabConditionSB.Text
    End Sub

    Private Sub cmbWearingSurfaceConditionSB_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmbWearingSurfaceConditionSB.SelectedIndexChanged
        strWearingSurfaceConditionSB = cmbWearingSurfaceConditionSB.Text
    End Sub

    Private Sub cmbInspectionSB_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmbInspectionSB.SelectedIndexChanged
        strInspectionSB = cmbInspectionSB.Text
    End Sub

    Private Sub cmbRedundancySB_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmbRedundancySB.SelectedIndexChanged
        strRedundancySB = cmbRedundancySB.Text
    End Sub

    Private Sub cmbMaintenanceSB_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmbMaintenanceSB.SelectedIndexChanged
        strMaintenanceSB = cmbMaintenanceSB.Text
    End Sub

    'Concrete grade - dblCG
    Private Sub txtConcreteGrade_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtConcreteGradeSB.TextChanged
        Try
            dblCGSB = txtConcreteGradeSB.Text
        Catch ex As Exception
            If txtConcreteGradeSB.Text = String.Empty Or txtConcreteGradeSB.Text = "." Then
                dblCGSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Concrete Grade'", "Data Error")
                txtConcreteGradeSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Diameter of middle strip bar - dblDbm
    Private Sub txtDiameterofMiddleStripBar_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtDiameterofMiddleStripBarSB.TextChanged
        Try
            dblDbmSB = txtDiameterofMiddleStripBarSB.Text
        Catch ex As Exception
            If txtDiameterofMiddleStripBarSB.Text = String.Empty Or
txtDiameterofMiddleStripBarSB.Text = "." Then
```

```
                dblDbmSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Diameter of Bar'", "Data Error")
                txtDiameterofMiddleStripBarSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Spacing of middle strip bar - dblSbm
    Private Sub txtSpacingOfMiddleStripBar_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtSpacingOfMiddleStripBarSB.TextChanged
        Try
            dblSbmSB = txtSpacingOfMiddleStripBarSB.Text
        Catch ex As Exception
            If txtSpacingOfMiddleStripBarSB.Text = String.Empty Or
txtSpacingOfMiddleStripBarSB.Text = "." Then
                dblSbmSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Spacing of Bar'", "Data Error")
                txtSpacingOfMiddleStripBarSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Yield stress of middle strip bars - dblYbm
    Private Sub txtYieldStressOfMiddleStripBar_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtYieldStressOfMiddleStripBarSB.TextChanged
        Try
            dblYbmSB = txtYieldStressOfMiddleStripBarSB.Text
        Catch ex As Exception
            If txtYieldStressOfMiddleStripBarSB.Text = String.Empty Or
txtYieldStressOfMiddleStripBarSB.Text = "." Then
                dblYbmSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Yield Stress of Bar'", "Data
Error")
                txtYieldStressOfMiddleStripBarSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Diameter of edge strip bar - dblDbe
    Private Sub txtDiameterofEdgeStripBar_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtDiameterofEdgeStripBarSB.TextChanged
        Try
            dblDbeSB = txtDiameterofEdgeStripBarSB.Text
        Catch ex As Exception
            If txtDiameterofEdgeStripBarSB.Text = String.Empty Or txtDiameterofEdgeStripBarSB.Text
= "." Then
                dblDbeSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Diameter of Bar'", "Data Error")
                txtDiameterofEdgeStripBarSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Spacing of edge strip bar - dblSbe
    Private Sub txtSpacingOfEdgeStripBar_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtSpacingOfEdgeStripBarSB.TextChanged
        Try
            dblSbeSB = txtSpacingOfEdgeStripBarSB.Text
        Catch ex As Exception
            If txtSpacingOfEdgeStripBarSB.Text = String.Empty Or txtSpacingOfEdgeStripBarSB.Text =
"." Then
```

```
                dblSbeSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Spacing of Bar'", "Data Error")
                txtSpacingOfEdgeStripBarSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Yield stress of edge strip bars - dblYbe
    Private Sub txtYieldStressOfEdgeStripBar_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtYieldStressOfEdgeStripBarSB.TextChanged
        Try
            dblYbeSB = txtYieldStressOfEdgeStripBarSB.Text
        Catch ex As Exception
            If txtYieldStressOfEdgeStripBarSB.Text = String.Empty Or
txtYieldStressOfEdgeStripBarSB.Text = "." Then
                dblYbeSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Yield Stress of Bar'", "Data
Error")
                txtYieldStressOfEdgeStripBarSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Number of lanes - dblNL
    Private Sub txtNumberOfLanes_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtNumberOfLanesSB.TextChanged
        Try
            intNLSB = txtNumberOfLanesSB.Text
            If txtNumberOfLanesSB.Text.Contains(".") Then
                MessageBox.Show("Please give integer value only", "Data Error")
                txtNumberOfLanesSB.Text = String.Empty
            End If
        Catch ex As Exception
            If txtNumberOfLanesSB.Text = String.Empty Or txtNumberOfLanesSB.Text = "." Then
                intNLSB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Number of Lanes'", "Data Error")
                txtNumberOfLanesSB.Text = String.Empty
            End If
        End Try
    End Sub

    'Check for whether wearing surface thickness is measured or nominal
    Private Sub rdbMeasuredThickness_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles rdbMeasuredThicknessSB.CheckedChanged
        Try
            If rdbMeasuredThicknessSB.Checked = True Then
                dblDwsSB = txtWearingSurfaceThicknessSB.Text
            Else
                dblDwsSB = 1.2 * txtWearingSurfaceThicknessSB.Text
            End If
        Catch ex As Exception
        End Try
    End Sub

    'Check for whether wearing surface thickness is measured or nominal
    Private Sub rdbNominalThickness_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles rdbNominalThicknessSB.Click
        Try
            If rdbNominalThicknessSB.Checked = False Then
                dblDwsSB = txtWearingSurfaceThicknessSB.Text
            Else
                dblDwsSB = 1.2 * txtWearingSurfaceThicknessSB.Text
```

```
            End If
        Catch ex As Exception
        End Try
    End Sub

    'DisplayInputValuesSB - method use to display calculated values
    Public Sub DisplayInputValuesSB()
        txtResultsSB.AppendText(Environment.NewLine & "SLAB BRIDGE STRENGTH EVALUATION")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "INPUT VALUES")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "GENERAL INFORMATIONS")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Bridge Identification Number:    " &
strBridgeIdentificationNumberSB)
        txtResultsSB.AppendText(Environment.NewLine & "Bridge Name:    " & strBridgeNameSB)
        txtResultsSB.AppendText(Environment.NewLine & "River Name:    " & strRiverNameSB)
        txtResultsSB.AppendText(Environment.NewLine & "Railing Material Type:    " & strRailingSB)
        txtResultsSB.AppendText(Environment.NewLine & "Post Material Type:    " & strPostSB)
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "DIMENTSIONS")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Total Length of Superstructure   =   " &
dblLtSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Span Length   =   " & dblSSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Total Width of Superstructure   =   " &
dblWtSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Clear Width of Supersturcture   =   " &
dblWcSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Curb/Edge Beam Width (top)   =   " &
dblWetSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Curb/Edge Beam Width (bottom)   =   " &
dblWebSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Additianal Curb/Edge Beam Depth   =   " &
dblDeSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Wearing Surface Thickness   =   " &
dblDwsSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Slab Thickness   =   " & dblDsSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Railing Depth   =   " & dblDrSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Railing Width   =   " & dblWrSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Post Depth   =   " & dblDpSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Post Width   =   " & dblWpSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Post Height   =   " & dblHpSB & " m")
        txtResultsSB.AppendText(Environment.NewLine & "Number of Lanes   =   " & intNLSB)
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "BRIDGE CONDITION")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Traffic Condition:   " &
strTrafficConditionSB)
        txtResultsSB.AppendText(Environment.NewLine & "Edge Strip Slab Condition:   " &
strEdgeStripSlabConditionSB)
        txtResultsSB.AppendText(Environment.NewLine & "Middle Strip Slab Condition:   " &
strMiddleStripSlabConditionSB)
        txtResultsSB.AppendText(Environment.NewLine & "Wearing Surface Condition:   " &
strWearingSurfaceConditionSB)
        txtResultsSB.AppendText(Environment.NewLine & "Inspection:   " & strInspectionSB)
        txtResultsSB.AppendText(Environment.NewLine & "Redundancy:   " & strRedundancySB)
        txtResultsSB.AppendText(Environment.NewLine & "Maintenance:   " & strMaintenanceSB)
        txtResultsSB.AppendText(Environment.NewLine & "Cube Strength of Concrete   =   " &
dblCGSB)
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "REINFORCEMENT")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Middle Strip Reinforcement")
        txtResultsSB.AppendText(Environment.NewLine & "")
```

```
        txtResultsSB.AppendText(Environment.NewLine & "Diameter   =   " & dblDbmSB & " mm")
        txtResultsSB.AppendText(Environment.NewLine & "Spacing   =   " & dblSbmSB & " mm")
        txtResultsSB.AppendText(Environment.NewLine & "Yield Stress   =   " & dblYbmSB & " MPa")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Edge Strip Reinfrocement")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Diameter   =   " & dblDbeSB & " mm")
        txtResultsSB.AppendText(Environment.NewLine & "Spacing   =   " & dblSbeSB & " mm")
        txtResultsSB.AppendText(Environment.NewLine & "Yield Stress   =   " & dblYbeSB & " MPa")
    End Sub
    'Variable Declaration - Load, strength and impact factor
    Dim dblGamaDLSB As Double = 1.2  'Deal load factor
    Dim dblGamaLLSB As Double 'Live load factor
    Dim dblPhieSB As Double 'Strength reduction factor
    Dim dblPhiiSB As Double 'strength reduction factor for middle strip
    Dim dblIFSB As Double 'Impact factor

    'Variable Declaration - Material unit weight in KN/m3
    Dim dblGamaC As Double = 24.0 'Unit weight of concrete
    Dim dblGamaWS As Double = 22.5 'Unit weight of wearing surface
    Dim dblGamaS As Double = 79.0 'Unit weight of steel
    Dim dblGamaT As Double = 8.0 'Unit weight of timber
    Dim dblGamaRSB As Double 'Unit weigth for railing
    Dim dblGamaPSB As Double 'Unit weight for post

    'Equivalent strip width
    Dim dblEmSB, dblEeSB As Double

    'Dead load
    Dim dblDLmSB, dblDLeSB As Double

    'Live load
    Dim dblPmSB, dblPeSB As Double

    'Variable Declaration - Array for shear force and moment
    'Dead load
    Dim dblVdiSB(10), dblMdiSB(10), dblVdeSB(10), dblMdeSB(10) As Double
    'Middle and edge strip- truck type 3
    Dim dblV3iSB(11), dblM3iSB(11), dblV3eSB(11), dblM3eSB(11) As Double
    'Middle and edge strip - truck type 3-2
    Dim dblV3_2iSB(11), dblM3_2iSB(11), dblV3_2eSB(11), dblM3_2eSB(11) As Double
    'Middle and edge strip - truck type 3-3
    Dim dblV3_3iSB(11), dblM3_3iSB(11), dblV3_3eSB(11), dblM3_3eSB(11) As Double
    'Middle and edge strip 'LLL - legal lane loading
    Dim dblVLLiSB(11), dblMLLiSB(11), dblVLLeSB(11), dblMLLeSB(11) As Double 'Live + dead load
shear force

    'Maximum shear force and moment for each case
    Dim dblMax_V3iSB, dblMax_V3_2iSB, dblMax_V3_3iSB, dblMax_VLLiSB As Double 'Middle Strip- shear
    Dim dblMax_M3iSB, dblMax_M3_2iSB, dblMax_M3_3iSB, dblMax_MLLiSB As Double 'Middle strip -
moment
    Dim dblMax_V3eSB, dblMax_V3_2eSB, dblMax_V3_3eSB, dblMax_VLLeSB As Double 'Edge strip - shear
    Dim dblMax_M3eSB, dblMax_M3_2eSB, dblMax_M3_3eSB, dblMax_MLLeSB As Double 'Edge strip - moment

    Dim dblViSB, dblMiSB As Double 'Maximum factored shear force and moment
    Dim dblVriSB, dblMriSB, dblVreSB, dblMreSB As Double 'Factored shear and moment resistance
    Dim dblVrdiSB, dblMrdiSB, dblVrdeSB, dblMrdeSB As Double 'Factored shear and moment resistance
minus dead load effect

    'Rating Factors
    Dim dblRFV3iSB, dblRFV3_2iSB, dblRFV3_3iSB, dblRFVLLiSB, dblRFM3iSB, dblRFM3_2iSB,
dblRFM3_3iSB, dblRFMLLiSB As Double
    Dim dblRFV3eSB, dblRFV3_2eSB, dblRFV3_3eSB, dblRFVLLeSB, dblRFM3eSB, dblRFM3_2eSB,
dblRFM3_3eSB, dblRFMLLeSB As Double
    Dim dblRFSB As Double
```

```
    'ImpactFactor - method to determine impact factor
    Public Sub ImpactFactor(ByVal strWearingSurfaceCondition As String, ByVal strBridge As String)
        Dim dblIF As Double
        Select Case strWearingSurfaceCondition
            Case "Good Condition"
                dblIF = 0.1
            Case "Fair Condition"
                dblIF = 0.1
            Case "Poor Condition"
                dblIF = 0.2
            Case "Critical Condition"
                dblIF = 0.3
        End Select

        Select Case strBridge
            Case "Slab"
                dblIFSB = dblIF
            Case "Girder"
                dblIFGB = dblIF
        End Select
    End Sub


    'LiveLoadFactor - method to detrmine live load factor from traffic condition
    Public Sub LiveLoadFactor(ByVal strTrafficCondition As String, ByVal strBridge As String)
        Dim dblGamaLL As Double
        Select Case strTrafficCondition
            Case "Low volume roadways (ADTT less than 1000), reasonable enforcement and apparent
control of overloads"
                dblGamaLL = 1.3
            Case "Heavy volume roadways (ADTT greater than 1000), reasonable enforcement and
apparent control of overloads (not common in Ethiopia)"
                dblGamaLL = 1.45
            Case "Low volume roadways (ADTT less than 1000), significant sources of overloads
without effective enforcement (common in Ethiopia)"
                dblGamaLL = 1.65
            Case "Heavy volume roadways (ADTT greater than 1000), significant sources of overloads
without effective enforcement"
                dblGamaLL = 1.8
        End Select

        Select Case strBridge
            Case "Slab"
                dblGamaLLSB = dblGamaLL
            Case "Girder"
                dblgamaLLGB = dblGamaLL
        End Select
    End Sub

    'StrengthReductionFactor - method to determine the strength reduction factor (dblPhi)
    Public Sub StrengthReductionFactor(ByVal Redundancy As String, ByVal Condition As String,
ByVal Inspection As String, ByVal Maintenance As String, ByVal strBridge As String, ByVal
strLocation As String)
        Dim dblPhi As Double
        Select Case Redundancy
            Case "Yes"
                dblPhi = 0.9
            Case "No"
                dblPhi = 0.8
        End Select
        Select Case Condition
            Case "Good to Fair"
                dblPhi = dblPhi
            Case "Deteriorated"
```

```
            dblPhi = dblPhi - 0.1
            Select Case Inspection
                Case "Careful"
                    dblPhi = dblPhi + 0.05
                Case "Estimated"
                    dblPhi = dblPhi
            End Select
        Case "Heavily Deteriorated"
            dblPhi = dblPhi - 0.2
            Select Case Inspection
                Case "Careful"
                    dblPhi = dblPhi + 0.05
                Case "Estimated"
                    dblPhi = dblPhi
            End Select
    End Select
    Select Case Maintenance
        Case "Vigorous"
            dblPhi = dblPhi + 0.05
        Case "Intermittent"
            dblPhi = dblPhi - 0.05
    End Select
    If dblPhi > 0.95 Then
        dblPhi = 0.95
    Else
        dblPhi = dblPhi
    End If

    Select Case strBridge
        Case "Slab"
            Select Case strLocation
                Case "Interior"
                    dblPhiiSB = dblPhi
                Case "Exterior"
                    dblPhieSB = dblPhi
            End Select
        Case "Girder"
            Select Case strLocation
                Case "Interior"
                    dblPhiiGB = dblPhi
                Case "Exterior"
                    dblPhieGB = dblPhi
            End Select
    End Select
End Sub

'UnitWeight - method to determine unit weight
Public Sub UnitWeight(ByVal Railing As String, ByVal Post As String, ByVal Bridge As String)
    Dim dblgamaR, dblgamaP As Double
    Select Case Railing
        Case "Concrete"
            dblgamaR = 24
        Case "Steel"
            dblgamaR = 79
        Case "Timber"
            dblgamaR = 8
    End Select
    Select Case Post
        Case "Concrete"
            dblgamaP = 24
        Case "Steel"
            dblgamaP = 79
        Case "Timber"
            dblgamaP = 8
    End Select
```

```
        Select Case Bridge
            Case "Slab"
                dblGamaRSB = dblgamaR
                dblGamaPSB = dblgamaP
            Case "Girder"
                dblGamaRGB = dblgamaR
                dblGamaPGB = dblgamaP
        End Select
    End Sub

    'Dead load shear force and moment calculation
    Public Sub DeadLoadShearAndMoment(ByVal Load As Double, ByVal strBridge As String, ByVal
strLocation As String)
        Dim dblA3() As Double = {0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5}
        Dim dblVd(10), dblMd(10) As Double
        Dim dblS As Double
        If strBridge = "Slab" Then
            dblS = dblSSB
        ElseIf strBridge = "Girder" Then
            dblS = dblSGB
        End If

        For intI As Integer = 0 To 10
            dblVd(intI) = Load * dblS * (0.5 - dblA3(intI))
            dblMd(intI) = Load * dblA3(intI) * dblS ^ 2 * 0.5 * (1 - dblA3(intI))
        Next

        Select Case strBridge
            Case "Slab"
                Select Case strLocation
                    Case "Interior"
                        dblVdiSB = dblVd
                        dblMdiSB = dblMd
                    Case "Exterior"
                        dblVdeSB = dblVd
                        dblMdeSB = dblMd
                End Select
            Case "Girder"
                Select Case strLocation
                    Case "Interior"
                        dblVdiGB = dblVd
                        dblMdiGB = dblMd
                    Case "Exterior"
                        dblVdeGB = dblVd
                        dblMdeGB = dblMd
                End Select
        End Select
    End Sub

    'TruckType3 -  method for analysis for truck type 3
    Public Sub TruckType3(ByVal Live As Double, ByVal BridgeType As String, ByVal
InteriorOrExterior As String, ByVal ShearOrMoment As String)

        'Variable declaration
        Dim dblS, dblGamaLL As Double
        If BridgeType = "Slab" Then
            dblS = dblSSB
            dblGamaLL = dblGamaLLSB
        ElseIf BridgeType = "Girder" Then
            dblS = dblSGB
            dblGamaLL = dblgamaLLGB
        End If
```

```
        Dim dblA3() As Double = {0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, (dblS /
2 - 0.52) / dblS}

        'Variable Declaration - Middle Strip
        Dim dblV3am(11) As Double 'Shear force at rear wheel position
        Dim dblV3bm(11) As Double 'Shear force at second wheel position
        Dim dblM3am(11) As Double 'Moment at rear wheel position
        Dim dblM3bm(11) As Double 'Moment at second wheel position

        'Variable Declaration - Edge Strip
        Dim dblV3ae(11) As Double 'Shear force at rear wheel position
        Dim dblV3be(11) As Double 'Shear force at second wheel position
        Dim dblM3ae(11) As Double 'Moment at rear wheel position
        Dim dblM3be(11) As Double 'Moment at second wheel position

        'Variable Declaration for x values
        Dim dblX1a(11) As Double
        Dim dblX2a(11) As Double
        Dim dblX3a(11) As Double
        Dim dblX1b(11) As Double
        Dim dblX2b(11) As Double
        Dim dblX3b(11) As Double

        'Case a) Wheel at rear position (truck moving to the right)
        'Populate the array for X1a
        For intI As Integer = 0 To 11
            dblX1a(intI) = dblS - dblA3(intI) * dblS - 5.7
            Select Case dblX1a(intI)
                Case Is < 0
                    dblX1a(intI) = 0
                Case Is > dblS
                    dblX1a(intI) = 0
            End Select
        Next

        'Populate the array for X2a
        For intI As Integer = 0 To 11
            dblX2a(intI) = dblS - dblA3(intI) * dblS - 1.2
            Select Case dblX2a(intI)
                Case Is < 0
                    dblX2a(intI) = 0
                Case Is > dblS
                    dblX2a(intI) = 0
            End Select
        Next

        'Populate the array for X3a
        For intI As Integer = 0 To 11
            dblX3a(intI) = dblS - dblA3(intI) * dblS
            Select Case dblX3a(intI)
                Case Is < 0
                    dblX3a(intI) = 0
                Case Is > dblS
                    dblX3a(intI) = 0
            End Select
        Next

        'Calculate live load shear and moment for case a
        Select Case InteriorOrExterior
            'Middle strip
            Case "Interior"
                For intI As Integer = 0 To 11
                    dblV3am(intI) = dblGamaLL * (Live / dblS * (0.73 * dblX1a(intI) + 0.77 *
dblX2a(intI) + 0.77 * dblX3a(intI)))
```

```
                    dblM3am(intI) = dblGamaLL * (Live * dblA3(intI) * (0.73 * dblX1a(intI) + 0.77
* dblX2a(intI) + 0.77 * dblX3a(intI)))
                Next

                'Edge strip
            Case "Exterior"

                For intI As Integer = 0 To 11
                    dblV3ae(intI) = dblGamaLL * (Live / dblS * (0.73 * dblX1a(intI) + 0.77 *
dblX2a(intI) + 0.77 * dblX3a(intI)))
                    dblM3ae(intI) = dblGamaLL * (Live * dblA3(intI) * (0.73 * dblX1a(intI) + 0.77
* dblX2a(intI) + 0.77 * dblX3a(intI)))
                Next

        End Select

        'Wheel at second position (truck moving to the right)
        'Populate the array for X1b
        For intI As Integer = 0 To 11
            dblX1b(intI) = dblS - dblA3(intI) * dblS - 4.5
            Select Case dblX1b(intI)
                Case Is < 0
                    dblX1b(intI) = 0
                Case Is > dblS
                    dblX1b(intI) = 0
            End Select
        Next

        'Populate the array for X2b
        For intI As Integer = 0 To 11
            dblX2b(intI) = dblS - dblA3(intI) * dblS
            Select Case dblX2b(intI)
                Case Is < 0
                    dblX2b(intI) = 0
                Case Is > dblS
                    dblX2b(intI) = 0
            End Select
        Next

        'Populate the array for X3b
        For intI As Integer = 0 To 11
            dblX3b(intI) = dblS - dblA3(intI) * dblS + 1.2
            Select Case dblX3b(intI)
                Case Is < 0
                    dblX3b(intI) = 0
                Case Is > dblS
                    dblX3b(intI) = 0
            End Select
        Next

        'Calculate live load shear and moment for case b
        Select Case InteriorOrExterior
            'Middle strip
            Case "Interior"
                For intI As Integer = 0 To 11
                    If dblX3b(intI) = 0 Then
                        dblV3bm(intI) = dblGamaLL * ((Live / dblS) * (0.73 * dblX1b(intI) + 0.77 *
dblX2b(intI) + 0.77 * dblX3b(intI)))
                        dblM3bm(intI) = dblGamaLL * (Live * dblA3(intI) * (0.73 * dblX1b(intI) +
0.77 * dblX2b(intI) + 0.77 * dblX3b(intI)))
                    Else
                        dblV3bm(intI) = dblGamaLL * ((Live / dblS) * (0.73 * dblX1b(intI) + 0.77 *
dblX2b(intI) + 0.77 * dblX3b(intI)) - 0.77 * Live)
                        dblM3bm(intI) = dblGamaLL * (Live * dblA3(intI) * (0.73 * dblX1b(intI) +
0.77 * dblX2b(intI) + 0.77 * dblX3b(intI)) - 0.924 * Live)
```

```
                    End If
                Next

                'Edge strip
            Case "Exterior"
                For intI As Integer = 0 To 11
                    If dblX3b(intI) = 0 Then
                        dblV3be(intI) = dblGamaLL * ((Live / dblS) * (0.73 * dblX1b(intI) + 0.77 * dblX2b(intI) + 0.77 * dblX3b(intI)))
                        dblM3be(intI) = dblGamaLL * (Live * dblA3(intI) * (0.73 * dblX1b(intI) + 0.77 * dblX2b(intI) + 0.77 * dblX3b(intI)))
                    Else
                        dblV3be(intI) = dblGamaLL * ((Live / dblS) * (0.73 * dblX1b(intI) + 0.77 * dblX2b(intI) + 0.77 * dblX3b(intI)) - 0.77 * Live)
                        dblM3be(intI) = dblGamaLL * (Live * dblA3(intI) * (0.73 * dblX1b(intI) + 0.77 * dblX2b(intI) + 0.77 * dblX3b(intI)) - 0.924 * Live)
                    End If
                Next
        End Select

        'Maximum shear force and moment out of the two cases assigned to the global variables
        Select Case BridgeType
            Case "Slab"
                Select Case InteriorOrExterior
                    Case "Interior"
                        For intI As Integer = 0 To 11
                            dblV3iSB(intI) = Math.Max(dblV3am(intI), dblV3bm(intI))
                            dblM3iSB(intI) = Math.Max(dblM3am(intI), dblM3bm(intI))
                        Next
                    Case "Exterior"
                        For intI As Integer = 0 To 11
                            dblV3eSB(intI) = Math.Max(dblV3ae(intI), dblV3be(intI))
                            dblM3eSB(intI) = Math.Max(dblM3ae(intI), dblM3be(intI))
                        Next
                End Select
            Case "Girder"
                Select Case InteriorOrExterior
                    Case "Interior"
                        If ShearOrMoment = "Shear" Then
                            For intI As Integer = 0 To 11
                                dblV3iGB(intI) = Math.Max(dblV3am(intI), dblV3bm(intI))
                            Next
                        ElseIf ShearOrMoment = "Moment" Then
                            For intI As Integer = 0 To 11
                                dblM3iGB(intI) = Math.Max(dblM3am(intI), dblM3bm(intI))
                            Next
                        End If
                    Case "Exterior"
                        If ShearOrMoment = "Shear" Then
                            For intI As Integer = 0 To 11
                                dblV3eGB(intI) = Math.Max(dblV3ae(intI), dblV3be(intI))
                            Next
                        ElseIf ShearOrMoment = "Moment" Then
                            For intI As Integer = 0 To 11
                                dblM3eGB(intI) = Math.Max(dblM3ae(intI), dblM3be(intI))
                            Next
                        End If
                End Select
        End Select
    End Sub

    'TruckType3_2 -  method for analysis for truck type 3_2
    Public Sub TruckType3_2(ByVal Live As Double, ByVal BridgeType As String, ByVal InteriorOrExterior As String, ByVal ShearOrMoment As String)
```

```
        'Variable declaration
        Dim dblL, dblGamaLL As Double
        If BridgeType = "Slab" Then
            dblL = dblSSB
            dblGamaLL = dblGamaLLSB
        ElseIf BridgeType = "Girder" Then
            dblL = dblSGB
            dblGamaLL = dblgamaLLGB
        End If

        Dim dblA3_2() As Double = {0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, (dblL
/ 2 - 1.11) / dblL}

        'Variable Declaration - Middle Strip
        Dim dblV3_2am(11) As Double 'Shear force at rear wheel position
        Dim dblV3_2bm(11) As Double 'Shear force at third wheel position
        Dim dblM3_2am(11) As Double 'Moment at rear wheel position
        Dim dblM3_2bm(11) As Double 'Moment at third wheel position
        Dim dblM3_2Maxm(11) As Double ' Maximum moment force either case a or b

        'Variable Declaration - Edge Strip
        Dim dblV3_2ae(11) As Double 'Shear force at rear wheel position
        Dim dblV3_2be(11) As Double 'Shear force at third wheel position
        Dim dblM3_2ae(11) As Double 'Moment at rear wheel position
        Dim dblM3_2be(11) As Double 'Moment at third wheel position
        Dim dblM3_2Maxe(11) As Double ' Maximum moment force

        'Variable Declaration for x values
        Dim dblX1a(11) As Double
        Dim dblX2a(11) As Double
        Dim dblX3a(11) As Double
        Dim dblX4a(11) As Double
        Dim dblX5a(11) As Double
        Dim dblX1b(11) As Double
        Dim dblX2b(11) As Double
        Dim dblX3b(11) As Double
        Dim dblX4b(11) As Double
        Dim dblX5b(11) As Double

        'Case a) Wheel at rear position (truck moving to the right)
        'Populate the array for X1a
        For intI As Integer = 0 To 11
            dblX1a(intI) = dblL - dblA3_2(intI) * dblL - 12.3
            Select Case dblX1a(intI)
                Case Is < 0
                    dblX1a(intI) = 0
                Case Is > dblL
                    dblX1a(intI) = 0
            End Select
        Next

        'Populate the array for X2a
        For intI As Integer = 0 To 11
            dblX2a(intI) = dblL - dblA3_2(intI) * dblL - 9
            Select Case dblX2a(intI)
                Case Is < 0
                    dblX2a(intI) = 0
                Case Is > dblL
                    dblX2a(intI) = 0
            End Select
        Next

        'Populate the array for X3a
        For intI As Integer = 0 To 11
            dblX3a(intI) = dblL - dblA3_2(intI) * dblL - 7.8
```

```
            Select Case dblX3a(intI)
                Case Is < 0
                    dblX3a(intI) = 0
                Case Is > dblL
                    dblX3a(intI) = 0
            End Select
        Next

        'Populate the array for X4a
        For intI As Integer = 0 To 11
            dblX4a(intI) = dblL - dblA3_2(intI) * dblL - 1.2
            Select Case dblX4a(intI)
                Case Is < 0
                    dblX4a(intI) = 0
                Case Is > dblL
                    dblX4a(intI) = 0
            End Select
        Next

        'Populate the array for X5a
        For intI As Integer = 0 To 11
            dblX5a(intI) = dblL - dblA3_2(intI) * dblL
            Select Case dblX5a(intI)
                Case Is < 0
                    dblX5a(intI) = 0
                Case Is > dblL
                    dblX5a(intI) = 0
            End Select
        Next

        'Calculate live load shear and moment for case a
        Select Case InteriorOrExterior
            'Middle strip
            Case "Interior"
                For intI As Integer = 0 To 11
                    dblV3_2am(intI) = dblGamaLL * (Live / dblL * (0.45 * dblX1a(intI) + 0.7 *
dblX2a(intI) + 0.7 * dblX3a(intI) + 0.7 * dblX4a(intI) + 0.7 * dblX5a(intI)))
                    dblM3_2am(intI) = dblGamaLL * (Live * dblA3_2(intI) * (0.45 * dblX1a(intI) +
0.7 * dblX2a(intI) + 0.7 * dblX3a(intI) + 0.7 * dblX4a(intI) + 0.7 * dblX5a(intI)))
                Next

                'Edge strip
            Case "Exterior"
                For intI As Integer = 0 To 11
                    dblV3_2ae(intI) = dblGamaLL * (Live / dblL * (0.45 * dblX1a(intI) + 0.7 *
dblX2a(intI) + 0.7 * dblX3a(intI) + 0.7 * dblX4a(intI) + 0.7 * dblX5a(intI)))
                    dblM3_2ae(intI) = dblGamaLL * (Live * dblA3_2(intI) * (0.45 * dblX1a(intI) +
0.7 * dblX2a(intI) + 0.7 * dblX3a(intI) + 0.7 * dblX4a(intI) + 0.7 * dblX5a(intI)))
                Next
        End Select

        'Wheel at second position (truck moving to the left)
        'Populate the array for X1b3_2
        For intI As Integer = 0 To 11
            dblX1b(intI) = dblL - dblA3_2(intI) * dblL + 4.5
            Select Case dblX1b(intI)
                Case Is < 0
                    dblX1b(intI) = 0
                Case Is > dblL
                    dblX1b(intI) = 0
            End Select
            'lstOutPut.Items.Add("X1: " & dblX1(intI))
        Next

        'Populate the array for X2b
```

```
        For intI As Integer = 0 To 11
            dblX2b(intI) = dblL - dblA3_2(intI) * dblL + 1.2
            Select Case dblX2b(intI)
                Case Is < 0
                    dblX2b(intI) = 0
                Case Is > dblL
                    dblX2b(intI) = 0
            End Select
            'lstOutPut.Items.Add("x2: " & dblX2(intI))
        Next

        'Populate the array for X3b
        For intI As Integer = 0 To 11
            dblX3b(intI) = dblL - dblA3_2(intI) * dblL
            Select Case dblX3b(intI)
                Case Is < 0
                    dblX3b(intI) = 0
                Case Is > dblL
                    dblX3b(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next

        'Populate the array for X4b
        For intI As Integer = 0 To 11
            dblX4b(intI) = dblL - dblA3_2(intI) * dblL - 6.6
            Select Case dblX4b(intI)
                Case Is < 0
                    dblX4b(intI) = 0
                Case Is > dblL
                    dblX4b(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next

        'Populate the array for X5b
        For intI As Integer = 0 To 11
            dblX5b(intI) = dblL - dblA3_2(intI) * dblL - 7.8
            Select Case dblX5b(intI)
                Case Is < 0
                    dblX5b(intI) = 0
                Case Is > dblL
                    dblX5b(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next

        'Calculate live load shear and moment for case b
        Select Case InteriorOrExterior
            'Middle strip
            Case "Interior"
                For intI As Integer = 0 To 11
                    If dblX2b(intI) = 0 Then
                        dblV3_2bm(intI) = dblGamaLL * ((Live / dblL) * (0.45 * dblX1b(intI) + 0.7
* dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)))
                        dblM3_2bm(intI) = dblGamaLL * (Live * dblA3_2(intI) * (0.45 * dblX1b(intI)
+ 0.7 * dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)))
                    ElseIf dblX1b(intI) = 0 Then
                        dblV3_2bm(intI) = dblGamaLL * ((Live / dblL) * (0.45 * dblX1b(intI) + 0.7
* dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)) - 0.7 * Live)
                        dblM3_2bm(intI) = dblGamaLL * (Live * dblA3_2(intI) * (0.45 * dblX1b(intI)
+ 0.7 * dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)) - 0.84 *
Live)
                    Else
```

```
                        dblV3_2bm(intI) = dblGamaLL * ((Live / dblL) * (0.45 * dblX1b(intI) + 0.7
* dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)) - 1.15 * Live)
                        dblM3_2bm(intI) = dblGamaLL * (Live * dblA3_2(intI) * (0.45 * dblX1b(intI)
+ 0.7 * dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)) - 2.865 *
Live)
                    End If
                    'lstOutPut.Items.Add("V3LM: " & dblV3LM(intI) & "  dblM3LM: " & dblM3LM(intI))
                Next

                'Edge strip
            Case "Exterior"
                For intI As Integer = 0 To 11
                    If dblX2b(intI) = 0 Then
                        dblV3_2be(intI) = dblGamaLL * ((Live / dblL) * (0.45 * dblX1b(intI) + 0.7
* dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)))
                        dblM3_2be(intI) = dblGamaLL * (Live * dblA3_2(intI) * (0.45 * dblX1b(intI)
+ 0.7 * dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)))
                    ElseIf dblX1b(intI) = 0 Then
                        dblV3_2be(intI) = dblGamaLL * ((Live / dblL) * (0.45 * dblX1b(intI) + 0.7
* dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)) - 0.7 * Live)
                        dblM3_2be(intI) = dblGamaLL * (Live * dblA3_2(intI) * (0.45 * dblX1b(intI)
+ 0.7 * dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)) - 0.84 *
Live)
                    Else
                        dblV3_2be(intI) = dblGamaLL * ((Live / dblL) * (0.45 * dblX1b(intI) + 0.7
* dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)) - 1.15 * Live)
                        dblM3_2be(intI) = dblGamaLL * (Live * dblA3_2(intI) * (0.45 * dblX1b(intI)
+ 0.7 * dblX2b(intI) + 0.7 * dblX3b(intI) + 0.7 * dblX4b(intI) + 0.7 * dblX5b(intI)) - 2.865 *
Live)
                    End If
                    'lstOutPut.Items.Add("V3LM: " & dblV3LM(intI) & "  dblM3LM: " & dblM3LM(intI))
                Next
        End Select

        'Maximum shear force and moment out of the two cases assigned to the global variables
        Select Case BridgeType
            Case "Slab"
                Select Case InteriorOrExterior
                    Case "Interior"
                        For intI As Integer = 0 To 11
                            dblV3_2iSB(intI) = Math.Max(dblV3_2am(intI), dblV3_2bm(intI))
                            dblM3_2iSB(intI) = Math.Max(dblM3_2am(intI), dblM3_2bm(intI))
                        Next
                    Case "Exterior"
                        For intI As Integer = 0 To 11
                            dblV3_2eSB(intI) = Math.Max(dblV3_2ae(intI), dblV3_2be(intI))
                            dblM3_2eSB(intI) = Math.Max(dblM3_2ae(intI), dblM3_2be(intI))
                        Next
                End Select
            Case "Girder"
                Select Case InteriorOrExterior
                    Case "Interior"
                        If ShearOrMoment = "Shear" Then
                            For intI As Integer = 0 To 11
                                dblV3_2iGB(intI) = Math.Max(dblV3_2am(intI), dblV3_2bm(intI))
                            Next
                        ElseIf ShearOrMoment = "Moment" Then
                            For intI As Integer = 0 To 11
                                dblM3_2iGB(intI) = Math.Max(dblM3_2am(intI), dblM3_2bm(intI))
                            Next
                        End If
                    Case "Exterior"
                        If ShearOrMoment = "Shear" Then
                            For intI As Integer = 0 To 11
                                dblV3_2eGB(intI) = Math.Max(dblV3_2ae(intI), dblV3_2be(intI))
```

```
                        Next
                    ElseIf ShearOrMoment = "Moment" Then
                        For intI As Integer = 0 To 11
                            dblM3_2eGB(intI) = Math.Max(dblM3_2ae(intI), dblM3_2be(intI))
                        Next
                    End If
                End Select
        End Select
    End Sub

    'TruckType3_3 -  method for analysis for truck type 3_3
    Public Sub TruckType3_3(ByVal Live As Double, ByVal BridgeType As String, ByVal
InteriorOrExterior As String, ByVal ShearOrMoment As String)

        'Variable declaration
        Dim dblL, dblGamaLL As Double
        If BridgeType = "Slab" Then
            dblL = dblSSB
            dblGamaLL = dblGamaLLSB
        ElseIf BridgeType = "Girder" Then
            dblL = dblSGB
            dblGamaLL = dblgamaLLGB
        End If

        Dim dblA3_3() As Double = {0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, (dblL
/ 2 - 0.59) / dblL}

        'Variable Declaration - Middle Strip
        Dim dblV3_3am(11) As Double 'Shear force at rear wheel position
        Dim dblV3_3bm(11) As Double 'Shear force at fourth wheel position
        Dim dblM3_3am(11) As Double 'Moment at rear wheel position
        Dim dblM3_3bm(11) As Double 'Moment at fourth wheel position

        'Variable Declaration - Edge Strip
        Dim dblV3_3ae(11) As Double 'Shear force at rear wheel position
        Dim dblV3_3be(11) As Double 'Shear force at fourth wheel position
        Dim dblM3_3ae(11) As Double 'Moment at rear wheel position
        Dim dblM3_3be(11) As Double 'Moment at fourth wheel position

        'Variable Declaration for x values
        Dim dblX1a(11) As Double
        Dim dblX2a(11) As Double
        Dim dblX3a(11) As Double
        Dim dblX4a(11) As Double
        Dim dblX5a(11) As Double
        Dim dblX6a(11) As Double
        Dim dblX1b(11) As Double
        Dim dblX2b(11) As Double
        Dim dblX3b(11) As Double
        Dim dblX4b(11) As Double
        Dim dblX5b(11) As Double
        Dim dblX6b(11) As Double

        'Case a) Wheel at rear position (truck moving to the right)
        'Populate the array for X1a
        For intI As Integer = 0 To 11
            dblX1a(intI) = dblL - dblA3_3(intI) * dblL - 16.3
            Select Case dblX1a(intI)
                Case Is < 0
                    dblX1a(intI) = 0
                Case Is > dblL
                    dblX1a(intI) = 0
            End Select
            'lstOutPut.Items.Add("X1: " & dblX1(intI))
        Next
```

```
'Populate the array for X2a
For intI As Integer = 0 To 11
    dblX2a(intI) = dblL - dblA3_3(intI) * dblL - 11.8
    Select Case dblX2a(intI)
        Case Is < 0
            dblX2a(intI) = 0
        Case Is > dblL
            dblX2a(intI) = 0
    End Select
    'lstOutPut.Items.Add("x2: " & dblX2(intI))
Next

'Populate the array for X3a
For intI As Integer = 0 To 11
    dblX3a(intI) = dblL - dblA3_3(intI) * dblL - 10.6
    Select Case dblX3a(intI)
        Case Is < 0
            dblX3a(intI) = 0
        Case Is > dblL
            dblX3a(intI) = 0
    End Select
    'lstOutPut.Items.Add("X3: " & dblX3(intI))
Next

'Populate the array for X4a
For intI As Integer = 0 To 11
    dblX4a(intI) = dblL - dblA3_3(intI) * dblL - 6.1
    Select Case dblX4a(intI)
        Case Is < 0
            dblX4a(intI) = 0
        Case Is > dblL
            dblX4a(intI) = 0
    End Select
    'lstOutPut.Items.Add("X3: " & dblX3(intI))
Next

'Populate the array for X5a
For intI As Integer = 0 To 11
    dblX5a(intI) = dblL - dblA3_3(intI) * dblL - 1.2
    Select Case dblX5a(intI)
        Case Is < 0
            dblX5a(intI) = 0
        Case Is > dblL
            dblX5a(intI) = 0
    End Select
    'lstOutPut.Items.Add("X3: " & dblX3(intI))
Next

'Populate the array for X5a
For intI As Integer = 0 To 11
    dblX6a(intI) = dblL - dblA3_3(intI) * dblL
    Select Case dblX6a(intI)
        Case Is < 0
            dblX6a(intI) = 0
        Case Is > dblL
            dblX6a(intI) = 0
    End Select
    'lstOutPut.Items.Add("X3: " & dblX3(intI))
Next

'Calculate live load shear and moment for case a
Select Case InteriorOrExterior
    'Middle strip
    Case "Interior"
```

```
            For intI As Integer = 0 To 11
                dblV3_3am(intI) = dblGamaLL * (Live / dblL * (0.55 * (dblX1a(intI) +
dblX2a(intI) + dblX3a(intI)) + 0.73 * dblX4a(intI) + 0.63 * (dblX5a(intI) + dblX6a(intI))))
                dblM3_3am(intI) = dblGamaLL * (Live * dblA3_3(intI) * (0.55 * (dblX1a(intI) +
dblX2a(intI) + dblX3a(intI)) + 0.73 * dblX4a(intI) + 0.63 * (dblX5a(intI) + dblX6a(intI))))
            Next

            'Edge strip
        Case "Exterior"
            For intI As Integer = 0 To 11
                dblV3_3ae(intI) = dblGamaLL * (Live / dblL * (0.55 * (dblX1a(intI) +
dblX2a(intI) + dblX3a(intI)) + 0.73 * dblX4a(intI) + 0.63 * (dblX5a(intI) + dblX6a(intI))))
                dblM3_3ae(intI) = dblGamaLL * (Live * dblA3_3(intI) * (0.55 * (dblX1a(intI) +
dblX2a(intI) + dblX3a(intI)) + 0.73 * dblX4a(intI) + 0.63 * (dblX5a(intI) + dblX6a(intI))))
            Next

    End Select

    'Wheel at fourth wheel position (truck moving to the right)
    'Populate the array for X1b
    For intI As Integer = 0 To 11
        dblX1b(intI) = dblL - dblA3_3(intI) * dblL - 10.2
        Select Case dblX1b(intI)
            Case Is < 0
                dblX1b(intI) = 0
            Case Is > dblL
                dblX1b(intI) = 0
        End Select
        'lstOutPut.Items.Add("X1: " & dblX1b(intI))
    Next

    'Populate the array for X2b
    For intI As Integer = 0 To 11
        dblX2b(intI) = dblL - dblA3_3(intI) * dblL - 5.7
        Select Case dblX2b(intI)
            Case Is < 0
                dblX2b(intI) = 0
            Case Is > dblL
                dblX2b(intI) = 0
        End Select
        'lstOutPut.Items.Add("x2: " & dblX2(intI))
    Next

    'Populate the array for X3b
    For intI As Integer = 0 To 11
        dblX3b(intI) = dblL - dblA3_3(intI) * dblL - 4.5
        Select Case dblX3b(intI)
            Case Is < 0
                dblX3b(intI) = 0
            Case Is > dblL
                dblX3b(intI) = 0
        End Select
        'lstOutPut.Items.Add("X3: " & dblX3(intI))
    Next

    'Populate the array for X4b
    For intI As Integer = 0 To 11
        dblX4b(intI) = dblL - dblA3_3(intI) * dblL
        Select Case dblX4b(intI)
            Case Is < 0
                dblX4b(intI) = 0
            Case Is > dblL
                dblX4b(intI) = 0
        End Select
        'lstOutPut.Items.Add("X3: " & dblX3(intI))
```

```
        Next

        'Populate the array for X5b
        For intI As Integer = 0 To 11
            dblX5b(intI) = dblL - dblA3_3(intI) * dblL + 4.9
            Select Case dblX5b(intI)
                Case Is < 0
                    dblX5b(intI) = 0
                Case Is > dblL
                    dblX5b(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next

        'Populate the array for X6b
        For intI As Integer = 0 To 11
            dblX6b(intI) = dblL - dblA3_3(intI) * dblL + 6.1
            Select Case dblX6b(intI)
                Case Is < 0
                    dblX6b(intI) = 0
                Case Is > dblL
                    dblX6b(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next

        'Calculate live load shear and moment for case b
        Select Case InteriorOrExterior
            'Middle strip
            Case "Interior"
                For intI As Integer = 0 To 11
                    If dblX5b(intI) = 0 Then
                        dblV3_3bm(intI) = dblGamaLL * (Live / dblL * (0.55 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) + dblX6b(intI))))
                        dblM3_3bm(intI) = dblGamaLL * (Live * dblA3_3(intI) * (0.55 *
(dblX1b(intI) + dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) +
dblX6b(intI))))
                    ElseIf dblX6b(intI) = 0 Then
                        dblV3_3bm(intI) = dblGamaLL * (Live / dblL * (0.55 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) + dblX6b(intI))) - 0.63
* Live)
                        dblM3_3bm(intI) = dblGamaLL * (Live * dblA3_3(intI) * (0.55 *
(dblX1b(intI) + dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) +
dblX6b(intI))) - 3.087 * Live)
                    Else
                        dblV3_3bm(intI) = dblGamaLLSB * (Live / dblL * (0.55 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) + dblX6b(intI))) - 1.26
* Live)
                        dblM3_3bm(intI) = dblGamaLLSB * (Live * dblA3_3(intI) * (0.55 *
(dblX1b(intI) + dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) +
dblX6b(intI))) - 6.93 * Live)
                    End If
                    'lstOutPut.Items.Add("V3LM: " & dblV3LM(intI) & "  dblM3LM: " & dblM3LM(intI))
                Next

                'Edge strip
            Case "Exterior"
                For intI As Integer = 0 To 11
                    If dblX5b(intI) = 0 Then
                        dblV3_3be(intI) = dblGamaLL * (Live / dblL * (0.55 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) + dblX6b(intI))))
                        dblM3_3be(intI) = dblGamaLL * (Live * dblA3_3(intI) * (0.55 *
(dblX1b(intI) + dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) +
dblX6b(intI))))
                    ElseIf dblX6b(intI) = 0 Then
```

```
                        dblV3_3be(intI) = dblGamaLL * (Live / dblL * (0.55 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) + dblX6b(intI))) - 0.63
* Live)
                        dblM3_3be(intI) = dblGamaLL * (Live * dblA3_3(intI) * (0.55 *
(dblX1b(intI) + dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) +
dblX6b(intI))) - 3.087 * Live)
                    Else
                        dblV3_3be(intI) = dblGamaLL * (Live / dblL * (0.55 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) + dblX6b(intI))) - 1.26
* Live)
                        dblM3_3be(intI) = dblGamaLL * (Live * dblA3_3(intI) * (0.55 *
(dblX1b(intI) + dblX2b(intI) + dblX3b(intI)) + 0.73 * dblX4b(intI) + 0.63 * (dblX5b(intI) +
dblX6b(intI))) - 6.93 * Live)
                    End If
                    'lstOutPut.Items.Add("V3LM: " & dblV3LM(intI) & "  dblM3LM: " & dblM3LM(intI))
                Next
        End Select

        'Maximum shear force and moment out of the two cases assigned to the global variables
        Select Case BridgeType
            Case "Slab"
                Select Case InteriorOrExterior
                    Case "Interior"
                        For intI As Integer = 0 To 11
                            dblV3_3iSB(intI) = Math.Max(dblV3_3am(intI), dblV3_3bm(intI))
                            dblM3_3iSB(intI) = Math.Max(dblM3_3am(intI), dblM3_3bm(intI))
                        Next
                    Case "Exterior"
                        For intI As Integer = 0 To 11
                            dblV3_3eSB(intI) = Math.Max(dblV3_3ae(intI), dblV3_3be(intI))
                            dblM3_3eSB(intI) = Math.Max(dblM3_3ae(intI), dblM3_3be(intI))
                        Next
                End Select
            Case "Girder"
                Select Case InteriorOrExterior
                    Case "Interior"
                        If ShearOrMoment = "Shear" Then
                            For intI As Integer = 0 To 11
                                dblV3_3iGB(intI) = Math.Max(dblV3_3am(intI), dblV3_3bm(intI))
                            Next
                        ElseIf ShearOrMoment = "Moment" Then
                            For intI As Integer = 0 To 11
                                dblM3_3iGB(intI) = Math.Max(dblM3_3am(intI), dblM3_3bm(intI))
                            Next
                        End If
                    Case "Exterior"
                        If ShearOrMoment = "Shear" Then
                            For intI As Integer = 0 To 11
                                dblV3_3eGB(intI) = Math.Max(dblV3_3ae(intI), dblV3_3be(intI))
                            Next
                        ElseIf ShearOrMoment = "Moment" Then
                            For intI As Integer = 0 To 11
                                dblM3_3eGB(intI) = Math.Max(dblM3_3ae(intI), dblM3_3be(intI))
                            Next
                        End If
                End Select
        End Select
    End Sub

    'TruckTypeLL -  method for analysis for truck type LL
    Public Sub TruckTypeLL(ByVal Live As Double, ByVal BridgeType As String, ByVal
InteriorOrExterior As String, ByVal ShearOrMoment As String)

        'Variable declaration
        Dim dblL, dblGamaLL As Double
```

```
        If BridgeType = "Slab" Then
            dblL = dblSSB
            dblGamaLL = dblGamaLLSB
        ElseIf BridgeType = "Girder" Then
            dblL = dblSGB
            dblGamaLL = dblgamaLLGB
        End If

        Dim dblAL() As Double = {0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, (dblL /
2 - 0.58) / dblL}

        'Variable Declaration - Middle Strip
        Dim dblVLLam(11) As Double 'Shear force at rear wheel position
        Dim dblVLLbm(11) As Double 'Shear force at fourth wheel position
        Dim dblMLLam(11) As Double 'Moment at rear wheel position
        Dim dblMLLbm(11) As Double 'Moment at fourth wheel position

        'Variable Declaration - Edge Strip
        Dim dblVLLae(11) As Double 'Shear force at rear wheel position
        Dim dblVLLbe(11) As Double 'Shear force at fourth wheel position
        Dim dblMLLae(11) As Double 'Moment at rear wheel position
        Dim dblMLLbe(11) As Double 'Moment at fourth wheel position

        'Variable Declaration for x values
        Dim dblX1a(11) As Double
        Dim dblX2a(11) As Double
        Dim dblX3a(11) As Double
        Dim dblX4a(11) As Double
        Dim dblX5a(11) As Double
        Dim dblX6a(11) As Double
        Dim dblX1b(11) As Double
        Dim dblX2b(11) As Double
        Dim dblX3b(11) As Double
        Dim dblX4b(11) As Double
        Dim dblX5b(11) As Double
        Dim dblX6b(11) As Double

        'Case a) Wheel at rear position (truck moving to the right)
        'Populate the array for X1a
        For intI As Integer = 0 To 11
            dblX1a(intI) = dblL - dblAL(intI) * dblL - 16.2
            Select Case dblX1a(intI)
                Case Is < 0
                    dblX1a(intI) = 0
                Case Is > dblL
                    dblX1a(intI) = 0
            End Select
            'lstOutPut.Items.Add("X1: " & dblX1(intI))
        Next

        'Populate the array for X2a
        For intI As Integer = 0 To 11
            dblX2a(intI) = dblL - dblAL(intI) * dblL - 11.7
            Select Case dblX2a(intI)
                Case Is < 0
                    dblX2a(intI) = 0
                Case Is > dblL
                    dblX2a(intI) = 0
            End Select
            'lstOutPut.Items.Add("x2: " & dblX2(intI))
        Next

        'Populate the array for X3a
        For intI As Integer = 0 To 11
            dblX3a(intI) = dblL - dblAL(intI) * dblL - 10.5
```

```
            Select Case dblX3a(intI)
                Case Is < 0
                    dblX3a(intI) = 0
                Case Is > dblL
                    dblX3a(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next

        'Populate the array for X4a
        For intI As Integer = 0 To 11
            dblX4a(intI) = dblL - dblAL(intI) * dblL - 6.0
            Select Case dblX4a(intI)
                Case Is < 0
                    dblX4a(intI) = 0
                Case Is > dblL
                    dblX4a(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next

        'Populate the array for X5a
        For intI As Integer = 0 To 11
            dblX5a(intI) = dblL - dblAL(intI) * dblL - 1.2
            Select Case dblX5a(intI)
                Case Is < 0
                    dblX5a(intI) = 0
                Case Is > dblL
                    dblX5a(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next

        'Populate the array for X6a
        For intI As Integer = 0 To 11
            dblX6a(intI) = dblL - dblAL(intI) * dblL
            Select Case dblX6a(intI)
                Case Is < 0
                    dblX6a(intI) = 0
                Case Is > dblL
                    dblX6a(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next
        Select Case InteriorOrExterior
            'Middle strip
            Case "Interior"
                For intI As Integer = 0 To 11
                    dblVLLam(intI) = dblGamaLL * (Live / dblL * (0.41 * (dblX1a(intI) +
dblX2a(intI) + dblX3a(intI)) + 0.54 * dblX4a(intI) + 0.48 * (dblX5a(intI) + dblX6a(intI)) + 0.015
* dblSSB ^ 2) - 0.03 * Live * dblAL(intI) * dblSSB)
                    dblMLLam(intI) = dblGamaLL * (Live * dblAL(intI) * (0.41 * (dblX1a(intI) +
dblX2a(intI) + dblX3a(intI)) + 0.54 * dblX4a(intI) + 0.48 * (dblX5a(intI) + dblX6a(intI)) + 0.015
* dblSSB ^ 2) - 0.015 * Live * dblAL(intI) ^ 2 * dblSSB ^ 2)
                Next

                'Edge strip
            Case "Exterior"
                For intI As Integer = 0 To 11
                    dblVLLae(intI) = dblGamaLL * (Live / dblL * (0.41 * (dblX1a(intI) +
dblX2a(intI) + dblX3a(intI)) + 0.54 * dblX4a(intI) + 0.48 * (dblX5a(intI) + dblX6a(intI)) + 0.015
* dblSSB ^ 2) - 0.03 * Live * dblAL(intI) * dblSSB)
                    dblMLLae(intI) = dblGamaLL * (Live * dblAL(intI) * (0.41 * (dblX1a(intI) +
dblX2a(intI) + dblX3a(intI)) + 0.54 * dblX4a(intI) + 0.48 * (dblX5a(intI) + dblX6a(intI)) + 0.015
* dblSSB ^ 2) - 0.015 * Live * dblAL(intI) ^ 2 * dblSSB ^ 2)
```

```
        Next
End Select

'Wheel at fourth position (truck moving to the right)
'Populate the array for X1b
For intI As Integer = 0 To 11
    dblX1b(intI) = dblL - dblAL(intI) * dblL - 10.2
    Select Case dblX1b(intI)
        Case Is < 0
            dblX1b(intI) = 0
        Case Is > dblL
            dblX1b(intI) = 0
    End Select
    'lstOutPut.Items.Add("X1: " & dblX1b(intI))
Next

'Populate the array for X2b
For intI As Integer = 0 To 11
    dblX2b(intI) = dblL - dblAL(intI) * dblL - 5.7
    Select Case dblX2b(intI)
        Case Is < 0
            dblX2b(intI) = 0
        Case Is > dblL
            dblX2b(intI) = 0
    End Select
    'lstOutPut.Items.Add("x2: " & dblX2(intI))
Next

'Populate the array for X3b
For intI As Integer = 0 To 11
    dblX3b(intI) = dblL - dblAL(intI) * dblL - 4.5
    Select Case dblX3b(intI)
        Case Is < 0
            dblX3b(intI) = 0
        Case Is > dblL
            dblX3b(intI) = 0
    End Select
    'lstOutPut.Items.Add("X3: " & dblX3(intI))
Next

'Populate the array for X4b
For intI As Integer = 0 To 11
    dblX4b(intI) = dblL - dblAL(intI) * dblL
    Select Case dblX4b(intI)
        Case Is < 0
            dblX4b(intI) = 0
        Case Is > dblL
            dblX4b(intI) = 0
    End Select
    'lstOutPut.Items.Add("X3: " & dblX3(intI))
Next

'Populate the array for X5b
For intI As Integer = 0 To 11
    dblX5b(intI) = dblL - dblAL(intI) * dblL + 4.8
    Select Case dblX5b(intI)
        Case Is < 0
            dblX5b(intI) = 0
        Case Is > dblL
            dblX5b(intI) = 0
    End Select
    'lstOutPut.Items.Add("X3: " & dblX3(intI))
Next

'Populate the array for X6b
```

```
        For intI As Integer = 0 To 11
            dblX6b(intI) = dblL - dblAL(intI) * dblL + 6.0
            Select Case dblX6b(intI)
                Case Is < 0
                    dblX6b(intI) = 0
                Case Is > dblL
                    dblX6b(intI) = 0
            End Select
            'lstOutPut.Items.Add("X3: " & dblX3(intI))
        Next

        'Calculate live load shear and moment for case b
        Select Case InteriorOrExterior
            'Middle strip
            Case "Interior"
                For intI As Integer = 0 To 11
                    If dblX5b(intI) = 0 Then
                        dblVLLbm(intI) = dblGamaLL * (Live / dblL * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.03 * Live * dblAL(intI) * dblSSB)
                        dblMLLbm(intI) = dblGamaLL * (Live * dblAL(intI) * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.015 * Live * dblAL(intI) ^ 2 * dblSSB ^ 2)
                    ElseIf dblX6b(intI) = 0 Then
                        dblVLLbm(intI) = dblGamaLL * (Live / dblL * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.03 * Live * dblAL(intI) * dblSSB - 0.48 * Live)
                        dblMLLbm(intI) = dblGamaLL * (Live * dblAL(intI) * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.015 * Live * dblAL(intI) ^ 2 * dblSSB ^ 2 - 2.304 * Live)
                    Else
                        dblVLLbm(intI) = dblGamaLL * (Live / dblL * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.03 * Live * dblAL(intI) * dblSSB - 0.96 * Live)
                        dblMLLbm(intI) = dblGamaLL * (Live * dblAL(intI) * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.015 * Live * dblAL(intI) ^ 2 * dblSSB ^ 2 - 5.184 * Live)
                    End If
                    'lstOutPut.Items.Add("V3LM: " & dblV3LM(intI) & "  dblM3LM: " & dblM3LM(intI))
                Next

                'Edge strip
            Case "Exterior"
                For intI As Integer = 0 To 11
                    If dblX5b(intI) = 0 Then
                        dblVLLbe(intI) = dblGamaLL * (Live / dblL * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.03 * Live * dblAL(intI) * dblSSB)
                        dblMLLbe(intI) = dblGamaLL * (Live * dblAL(intI) * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.015 * Live * dblAL(intI) ^ 2 * dblSSB ^ 2)
                    ElseIf dblX6b(intI) = 0 Then
                        dblVLLbe(intI) = dblGamaLL * (Live / dblL * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.03 * Live * dblAL(intI) * dblSSB - 0.48 * Live)
                        dblMLLbe(intI) = dblGamaLL * (Live * dblAL(intI) * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.015 * Live * dblAL(intI) ^ 2 * dblSSB ^ 2 - 2.304 * Live)
                    Else
                        dblVLLbe(intI) = dblGamaLL * (Live / dblL * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.03 * Live * dblAL(intI) * dblSSB - 0.96 * Live)
                        dblMLLbe(intI) = dblGamaLL * (Live * dblAL(intI) * (0.41 * (dblX1b(intI) +
dblX2b(intI) + dblX3b(intI)) + 0.54 * dblX4b(intI) + 0.48 * (dblX5b(intI) + dblX6b(intI)) + 0.015
* dblSSB ^ 2) - 0.015 * Live * dblAL(intI) ^ 2 * dblSSB ^ 2 - 5.184 * Live)
```

```
                End If
                'lstOutPut.Items.Add("V3LM: " & dblV3LM(intI) & "  dblM3LM: " & dblM3LM(intI))
            Next

    End Select

    'Maximum shear force and moment out of the two cases assigned to the global variables
    Select Case BridgeType
        Case "Slab"
            Select Case InteriorOrExterior
                Case "Interior"
                    For intI As Integer = 0 To 11
                        dblVLLiSB(intI) = Math.Max(dblVLLam(intI), dblVLLbm(intI))
                        dblMLLiSB(intI) = Math.Max(dblMLLam(intI), dblMLLbm(intI))
                    Next
                Case "Exterior"
                    For intI As Integer = 0 To 11
                        dblVLLeSB(intI) = Math.Max(dblVLLae(intI), dblVLLbe(intI))
                        dblMLLeSB(intI) = Math.Max(dblMLLae(intI), dblMLLbe(intI))
                    Next
            End Select
        Case "Girder"
            Select Case InteriorOrExterior
                Case "Interior"
                    If ShearOrMoment = "Shear" Then
                        For intI As Integer = 0 To 11
                            dblVLLiGB(intI) = Math.Max(dblVLLam(intI), dblVLLbm(intI))
                        Next
                    ElseIf ShearOrMoment = "Moment" Then
                        For intI As Integer = 0 To 11
                            dblMLLiGB(intI) = Math.Max(dblMLLam(intI), dblMLLbm(intI))
                        Next
                    End If
                Case "Exterior"
                    If ShearOrMoment = "Shear" Then
                        For intI As Integer = 0 To 11
                            dblVLLeGB(intI) = Math.Max(dblVLLae(intI), dblVLLbe(intI))
                        Next
                    ElseIf ShearOrMoment = "Moment" Then
                        For intI As Integer = 0 To 11
                            dblMLLeGB(intI) = Math.Max(dblMLLae(intI), dblMLLbe(intI))
                        Next
                    End If
            End Select
    End Select
End Sub

'Maximum - method to determint maximum load effect for each loading type
Public Function Maximum(ByVal value() As Double) As Double
    Dim dblMax As Double
    Dim dblJ As Double
    dblMax = value(0)
    For intI As Integer = 0 To 11
        If dblMax < value(intI) Then
            dblMax = value(intI)
            dblJ = intI
        End If
    Next
    Return dblMax
End Function

'ShearStrength - Method for calculating shear strength
Public Sub ShearStrengthSB(ByVal Strip As String)
    Dim dblVn As Double
    Dim dblD As Double = dblDsSB - 0.025
```

```
    dblVn = 0.083 * 2 * Math.Sqrt(0.8 * dblCGSB) * 1 * dblD * 1000.0
    If dblVn < 0.25 * 0.8 * dblCGSB * 1 * dblD * 1000.0 Then
        dblVn = dblVn
    Else
        dblVn = 0.25 * 0.8 * dblCGSB * 1 * dblD * 1000.0
    End If
    Select Case Strip
        Case "Middle"
            dblVriSB = dblPhiiSB * dblVn

        Case "Edge"
            dblVreSB = dblPhieSB * dblVn
    End Select

End Sub

'MomentStrength - Method for calculating shear strength
Public Sub MomentStrengthSB(ByVal Strip As String)
    If 0.8 * dblCGSB <= 28 Then : dblBSB = 0.85
    Else : dblBSB = 0.85 - 0.05 / 7 * (0.8 * dblCGSB - 28) : End If
    If dblBSB < 0.65 Then dblBSB = 0.65
    Dim dblMn As Double
    Dim dblD As Double = dblDsSB - 0.025
    Dim dblNbm As Double
    Dim dblNbe As Double
    Dim dblAs As Double
    Dim dblX As Double
    Select Case Strip
        Case "Middle"
            dblNbm = Math.Round(1000 / dblSbmSB + 1, 0)
            dblAs = dblNbm * Math.PI * (dblDbmSB * 0.001) ^ 2 / 4
            dblX = dblAs * dblYbmSB / (dblBSB * 0.85 * 0.8 * dblCGSB * 1)
            dblMn = dblAs * dblYbmSB * (dblD - dblBSB * dblX / 2) * 1000.0
            dblMriSB = dblPhiiSB * dblMn

        Case "Edge"
            dblNbe = Math.Round(1000 / dblSbeSB + 1, 0)
            dblAs = dblNbe * Math.PI * (dblDbeSB * 0.001) ^ 2 / 4
            dblX = dblAs * dblYbeSB / (dblBSB * 0.85 * 0.8 * dblCGSB * 1)
            dblMn = dblAs * dblYbeSB * (dblD - dblBSB * dblX / 2) * 1000.0
            dblMreSB = dblPhieSB * dblMn
    End Select
End Sub

'NumberOfLanes - method used to calculate Number of lanes
Public Sub NumberOfLanes(ByVal BridgeType As String)
    Select Case BridgeType
        Case "Slab"
            If txtNumberOfLanesSB.Text = String.Empty Then
                If txtClearWidthOfSuperstructureSB.Text <> String.Empty Then
                    intNLSB = Math.Floor(dblWcSB / 3)
                    If dblWcSB / 3 < 1 Then
                        intNLSB = 1
                    End If
                Else
                    intNLSB = 0
                End If
            End If
        Case "Girder"
            If txtNumberOfLanesGB.Text = String.Empty Then
                If txtClearWidthOfSuperstructureGB.Text <> String.Empty Then
                    intNLGB = Math.Floor(dblWcGB / 3)
                    If dblWcGB / 3 < 1 Then
                        intNLGB = 1
                    End If
```

```
                Else
                    intNLGB = 0
                End If
            End If
    End Select

End Sub

'EquivalentStripWidth - method used to calculate equivalent strip width
Public Sub EquivalentStripWidth(ByVal Strip As String)
    Select Case Strip
        Case "Interior"
            'One lane loaded
            Dim dblE1m As Double
            Dim dblL1 As Double
            Dim dblW11 As Double
            If dblSSB < 18 Then
                dblL1 = dblSSB
            Else
                dblL1 = 18
            End If
            If dblWtSB < 9 Then
                dblW11 = dblWtSB
            Else
                dblW11 = 9
            End If
            dblE1m = 0.001 * (250 + 0.42 * Math.Sqrt(dblL1 * dblW11 * 1000000.0))

            'More than one lane loaded
            Dim dblE2m As Double 'More than one lane loaded
            Dim dblW21 As Double
            If dblWtSB < 18 Then
                dblW21 = dblWtSB
            Else
                dblW21 = 18
            End If
            dblE2m = 0.001 * (2100 + 0.12 * Math.Sqrt(dblL1 * dblW21 * 1000000.0))
            If dblE2m > 3 Then
                dblE2m = 3
            Else
                dblE2m = dblE2m
            End If
            If dblE2m <= (dblWtSB / intNLSB) Then
                dblE2m = dblE2m
            Else
                dblE2m = (dblWtSB / intNLSB)
            End If

            If intNLSB < 2 Then
                dblEmSB = dblE1m
            Else
                If dblE1m <= dblE2m Then
                    dblEmSB = dblE1m
                Else
                    dblEmSB = dblE2m
                End If
            End If
        Case "Exterior"
            If dblDeSB = 0 Then
                dblEeSB = 0.5 * dblEmSB '+ 0.3
            Else
                dblEeSB = dblWebSB + 0.3 + 0.5 * dblEmSB
            End If

            If dblEeSB > dblEmSB Then
```

```
                    dblEeSB = dblEmSB
                ElseIf dblEeSB > 1.8 Then
                    dblEeSB = 1.8
                End If
        End Select
    End Sub
    'DeadLoadSB - method to calculate dead load of slab bridge
    Public Sub DeadLoadSB(ByVal Strip As String)
        Select Case Strip
            Case "Interior"
                dblDLmSB = dblGamaDLSB * (dblGamaC * dblDsSB + dblGamaWS * dblDwsSB)
            Case "Exterior"
                If dblSpSB <> 0 Then
                    dblNpSB = Math.Ceiling(dblLtSB / dblSpSB)
                Else
                    dblNpSB = 0
                End If

                Dim dblDLp As Double = dblGamaPSB * dblNpSB * dblWpSB * dblHpSB * dblDpSB
                Dim dblDLr As Double = dblGamaRSB * dblLtSB * dblWrSB * dblDrSB
                Dim dblDLc As Double = dblGamaC * dblLtSB * 0.5 * dblDeSB * (dblWetSB + dblWebSB)
                Dim dblDLprc As Double = (dblDLp + dblDLr + dblDLc) / (dblEeSB * dblLtSB)
                dblDLeSB = dblGamaDLSB * (dblGamaC * dblDsSB + dblGamaWS * dblDwsSB + dblDLprc)
        End Select
    End Sub

    'LiveLoadSB - method to calculate live load per axle
    Public Sub LiveLoad(ByVal Strip As String)
        Select Case Strip
            Case "Interior"
                dblPmSB = (100 / dblEmSB) * (1 + dblIFSB)
            Case "Exterior"
                dblPeSB = 50 / dblEeSB * (1 + dblIFSB)
        End Select
    End Sub
    'DisplayResultsGB - method used to display calculated values
    Public Sub DisplayResultsSB()
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "OUTPUT VALUES")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Number of lanes   =   " & intNLSB)
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "UNIT WEIGHTS")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Unit weight of Railing   =   " & dblGamaRSB
& " KN/m3")
        txtResultsSB.AppendText(Environment.NewLine & "Unit weight of Post   =   " & dblGamaPSB &
" KN/m3")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "FACTORS")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Dead Load Factor   =   " & dblGamaDLSB)
        txtResultsSB.AppendText(Environment.NewLine & "Live Load Factor   =   " & dblGamaLLSB)
        txtResultsSB.AppendText(Environment.NewLine & "Strength Reduction Factor - Edge Strip   =
" & dblPhieSB)
        txtResultsSB.AppendText(Environment.NewLine & "Strength Reduction Factor - Middle Strip
=   " & dblPhiiSB)
        txtResultsSB.AppendText(Environment.NewLine & "Impact Factor   =   " & dblIFSB)
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "DEAD LOADS")
        txtResultsSB.AppendText(Environment.NewLine & "Dead Load - Middle Strip   =   " &
Math.Round(dblDLmSB, 2) & " KN/m")
        txtResultsSB.AppendText(Environment.NewLine & "Dead Load - Edge Strip   =   " &
Math.Round(dblDLeSB, 2) & " KN/m")
        txtResultsSB.AppendText(Environment.NewLine & "")
```

```
        txtResultsSB.AppendText(Environment.NewLine & "DEAD LOAD SHEAR FORCE AND MOMENT RESULTS")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Middle Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "             " & "M (KN-m)")
        For intI As Integer = 0 To 10
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblVdiSB(intI), 2) & "
" & Math.Round(dblMdiSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Edge Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "             " & "M (KN-m)")
        For intI As Integer = 0 To 10
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblVdeSB(intI), 2) & "
" & Math.Round(dblMdeSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "EQUIVALENT STRIP WIDTH")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Equivalent Middle Strip Width   =   " &
dblEmSB)
        txtResultsSB.AppendText(Environment.NewLine & "Equivalent Edge Strip Width   =   " &
dblEeSB)
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "LIVE LOAD per AXEL")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Middle Strip (unfactored)   =   " &
Math.Round(dblPmSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Edge Strip (unfactored)   =   " &
Math.Round(dblPeSB, 2))
        'Live load shear force and moment
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "LIVE LOAD SHEAR FORCE (KN) AND MOMENT (KN-
m)")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Middle Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Due to truck type 3-1")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblV3iSB(intI), 2) & "
" & Math.Round(dblM3iSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Due to truck type 3_2")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblV3_2iSB(intI), 2) & "
" & Math.Round(dblM3_2iSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Due to truck type 3_3")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblV3_3iSB(intI), 2) & "
" & Math.Round(dblM3_3iSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Due to truck type LL")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
```

```
        For intI As Integer = 0 To 11
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblVLLiSB(intI), 2) & "
" & Math.Round(dblMLLiSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Edge Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Due to truck type 3-1")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblV3eSB(intI), 2) & "
" & Math.Round(dblM3eSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Due to truck type 3_2")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblV3_2eSB(intI), 2) & "
" & Math.Round(dblM3_2eSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Due to truck type 3_3")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblV3_3eSB(intI), 2) & "
" & Math.Round(dblM3_3eSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Due to truck type LL")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsSB.AppendText(Environment.NewLine & Math.Round(dblVLLeSB(intI), 2) & "
" & Math.Round(dblMLLeSB(intI), 2))
        Next
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "MAMIMUM SHEAR FORCE AND MOMENT FOR EACH
LOAD CASE")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Middle Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-1
=   " & Math.Round(dblMax_V3iSB, 2) & " KN")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-1   =
" & Math.Round(dblMax_M3iSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-2
=   " & Math.Round(dblMax_V3_2iSB, 2) & " KN")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-2   =
" & Math.Round(dblMax_M3_2iSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-3
=   " & Math.Round(dblMax_V3_3iSB, 2) & " KN")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-3 = " &
Math.Round(dblMax_M3_3iSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum shear force due to truck type legal
lane load   =   " & Math.Round(dblMax_VLLiSB, 2) & " KN")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum moment due to truck type legal lane
load   =   " & Math.Round(dblMax_MLLiSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Edge Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-1
=   " & Math.Round(dblMax_V3eSB, 2) & " KN")
```

```
        txtResultsSB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-1   =   " & Math.Round(dblMax_M3eSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-2   =   " & Math.Round(dblMax_V3_2eSB, 2) & " KN")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-2   =   " & Math.Round(dblMax_M3_2eSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-3   =   " & Math.Round(dblMax_V3_3eSB, 2) & " KN")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-3   =   " & Math.Round(dblMax_M3_3eSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum shear force due to truck type legal lane load   =   " & Math.Round(dblMax_VLLeSB, 2) & " KN")
        txtResultsSB.AppendText(Environment.NewLine & "Maximum moment due to truck type legal lane load   =   " & Math.Round(dblMax_MLLeSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "STRENGTH CALCULATION")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Middle Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Shear Strength   =   " & Math.Round(dblVriSB, 2) & " KN")
        txtResultsSB.AppendText(Environment.NewLine & "Moment Strength   =   " & Math.Round(dblMriSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Edge Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Shear strength   =   " & Math.Round(dblVreSB, 2) & " KN")
        txtResultsSB.AppendText(Environment.NewLine & "Moment capacity   =   " & Math.Round(dblMreSB, 2) & " KN-m")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "RATING FACTORS")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Middle Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Rating factor for shear")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3-1: RF   =   " & Math.Round(dblRFV3iSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3_2: RF   =   " & Math.Round(dblRFV3_2iSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3_3: RF   =   " & Math.Round(dblRFV3_3iSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Legal lane loading: RF   =   " & Math.Round(dblRFVLLiSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Rating factor for moment")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3-1: RF   =   " & Math.Round(dblRFM3iSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3_2: RF   =   " & Math.Round(dblRFM3_2iSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3_3: RF   =   " & Math.Round(dblRFM3_3iSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Legal lane loading: RF   =   " & Math.Round(dblRFMLLiSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Edge Strip")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Rating factor for shear")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3-1: R   =   " & Math.Round(dblRFV3eSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3_2: RF   =   " & Math.Round(dblRFV3_2eSB, 2))
```

```
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3_3: RF   =   " &
Math.Round(dblRFV3_3eSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Legal lane loading: RF   =   " &
Math.Round(dblRFVLLeSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Rating factor for moment")
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3-1: RF   =   " &
Math.Round(dblRFM3eSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3_2: RF   =   " &
Math.Round(dblRFM3_2eSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Truck Type 3_3: RF   =   " &
Math.Round(dblRFM3_3eSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "Legal lane loading: RF   =   " &
Math.Round(dblRFMLLeSB, 2))
        txtResultsSB.AppendText(Environment.NewLine & "")
        txtResultsSB.AppendText(Environment.NewLine & "RATING FACTOR   =   " & Math.Round(dblRFSB,
2))

    End Sub
    'Run - Method used to do the necessary calculations
    Public Sub RunSB()
        tbcSlab.SelectedTab = tbResultsSB
        txtResultsSB.Clear()
        DisplayInputValuesSB()

        NumberOfLanes("Slab")

        'Determination of unit weigth
        UnitWeight(strRailingSB, strPostSB, "Slab")

        'Deterination of impact factor
        ImpactFactor(strWearingSurfaceConditionSB, "Slab")

        'Determination of live load and strength reduction factors
        LiveLoadFactor(strTrafficConditionSB, "Slab")
        StrengthReductionFactor(strRedundancySB, strEdgeStripSlabConditionSB, strInspectionSB,
strMaintenanceSB, "Slab", "Exterior")
        StrengthReductionFactor(strRedundancySB, strMiddleStripSlabConditionSB, strInspectionSB,
strMaintenanceSB, "Slab", "Interior")
        'Determination of equivalent strip width
        EquivalentStripWidth("Interior")
        EquivalentStripWidth("Exterior")

        'Dead load calculation
        DeadLoadSB("Interior") : DeadLoadSB("Exterior")
        'Dead load shear force and moment calculation
        DeadLoadShearAndMoment(dblDLmSB, "Slab", "Interior")
        DeadLoadShearAndMoment(dblDLeSB, "Slab", "Exterior")

        'Live load
        LiveLoad("Interior") : LiveLoad("Exterior")

        'Analysis for truck type 3
        TruckType3(dblPmSB, "Slab", "Interior", "") : TruckType3(dblPeSB, "Slab", "Exterior", "")
        'Analysis for truck type 3_2
        TruckType3_2(dblPmSB, "Slab", "Interior", "") : TruckType3_2(dblPeSB, "Slab", "Exterior",
"")
        'Analysis for truck type 3_3
        TruckType3_3(dblPmSB, "Slab", "Interior", "") : TruckType3_3(dblPeSB, "Slab", "Exterior",
"")
        'Analysis for legal lane loading
        TruckTypeLL(dblPmSB, "Slab", "Interior", "") : TruckTypeLL(dblPeSB, "Slab", "Exterior",
"")
```

```
        'Get maximum values for each live load case
        'Shear force
        dblMax_V3iSB = Maximum(dblV3iSB) : dblMax_M3iSB = Maximum(dblM3iSB)
        dblMax_V3_2iSB = Maximum(dblV3_2iSB) : dblMax_M3_2iSB = Maximum(dblM3_2iSB)
        dblMax_V3_3iSB = Maximum(dblV3_3iSB) : dblMax_M3_3iSB = Maximum(dblM3_3iSB)
        dblMax_VLLiSB = Maximum(dblVLLiSB) : dblMax_MLLiSB = Maximum(dblMLLiSB)


        dblMax_V3eSB = Maximum(dblV3eSB) : dblMax_M3eSB = Maximum(dblM3eSB)
        dblMax_V3_2eSB = Maximum(dblV3_2eSB) : dblMax_M3_2eSB = Maximum(dblM3_2eSB)
        dblMax_V3_3eSB = Maximum(dblV3_3eSB) : dblMax_M3_3eSB = Maximum(dblM3_3eSB)
        dblMax_VLLeSB = Maximum(dblVLLeSB) : dblMax_MLLeSB = Maximum(dblMLLeSB)

        'Strength calculation
        ShearStrengthSB("Middle") : ShearStrengthSB("Edge")
        MomentStrengthSB("Middle") : MomentStrengthSB("Edge")

        'Rating Factor Calculation
        'Middle strip
        'Shear RF
        dblRFV3iSB = (dblVriSB - dblVdiSB(0)) / dblMax_V3iSB
        dblRFV3_2iSB = (dblVriSB - dblVdiSB(0)) / dblMax_V3_2iSB
        dblRFV3_3iSB = (dblVriSB - dblVdiSB(0)) / dblMax_V3_3iSB
        dblRFVLLiSB = (dblVriSB - dblVdiSB(0)) / dblMax_VLLiSB

        'Moment RF
        dblRFM3iSB = (dblMriSB - dblMdiSB(10)) / dblMax_M3iSB
        dblRFM3_2iSB = (dblMriSB - dblMdiSB(10)) / dblMax_M3_2iSB
        dblRFM3_3iSB = (dblMriSB - dblMdiSB(10)) / dblMax_M3_3iSB
        dblRFMLLiSB = (dblMriSB - dblMdiSB(10)) / dblMax_MLLiSB

        'Edge strip
        'Shear RF
        dblRFV3eSB = (dblVreSB - dblVdeSB(0)) / dblMax_V3eSB
        dblRFV3_2eSB = (dblVreSB - dblVdeSB(0)) / dblMax_V3_2eSB
        dblRFV3_3eSB = (dblVreSB - dblVdeSB(0)) / dblMax_V3_3eSB
        dblRFVLLeSB = (dblVreSB - dblVdeSB(0)) / dblMax_VLLeSB

        'Moment RF
        dblRFM3eSB = (dblMreSB - dblMdeSB(10)) / dblMax_M3eSB
        dblRFM3_2eSB = (dblMreSB - dblMdeSB(10)) / dblMax_M3_2eSB
        dblRFM3_3eSB = (dblMreSB - dblMdeSB(10)) / dblMax_M3_3eSB
        dblRFMLLeSB = (dblMreSB - dblMdeSB(10)) / dblMax_MLLeSB

        dblRFSB = Math.Min(Math.Min(Math.Min(Math.Min(dblRFV3iSB, dblRFV3eSB),
Math.Min(dblRFV3_2iSB, dblRFV3_2eSB)), Math.Min(Math.Min(dblRFV3_3iSB, dblRFV3_3eSB), _
Math.Min(dblRFVLLiSB, dblRFVLLeSB))), Math.Min(Math.Min(Math.Min(dblRFM3iSB, dblRFM3eSB),
Math.Min(dblRFM3_2iSB, dblRFM3_2eSB)), Math.Min(Math.Min(dblRFM3_3iSB, dblRFM3_3eSB), _
Math.Min(dblRFMLLiSB, dblRFMLLeSB))))

        DisplayResultsSB()
    End Sub

    Private Sub msClear_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
msClear.Click
        If tbcMain.SelectedTab.Equals(tbSlab) Then
            tbcSlab.SelectedTab = tbResultsSB
            txtResultsSB.Clear()
        ElseIf tbcMain.SelectedTab.Equals(tbGirder) Then
            tbcGirder.SelectedTab = tbResultsGB
            txtResultsGB.Clear()
        End If
    End Sub
```

```
    Private Sub SaveAsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SaveAsToolStripMenuItem.Click
        If tbcMain.SelectedTab.Equals(tbSlab) Then
            Dim saveFileDialog1 As New SaveFileDialog()
            saveFileDialog1.Filter = "Text Documents|*.txt"
            saveFileDialog1.Title = "Save the Result of Slab Bridge Strength Evaluation"
            saveFileDialog1.ShowDialog()
            If saveFileDialog1.FileName <> "" Then
                Dim sw As StreamWriter = New StreamWriter(saveFileDialog1.FileName)
                sw.Write(txtResultsSB.Text)
                sw.Close()
            End If
            saveFileDialog1.InitialDirectory = "C:\Documents and Settings\Administrator\My
Documents"
            'saveFileDialog1.AddExtension = True
        ElseIf tbcMain.SelectedTab.Equals(tbGirder) Then
            Dim saveFileDialog1 As New SaveFileDialog()
            saveFileDialog1.Filter = "Text Documents|*.txt"
            saveFileDialog1.Title = "Save the Result of Girder Bridge Strength Evaluation"
            saveFileDialog1.ShowDialog()
            If saveFileDialog1.FileName <> "" Then
                Dim sw As StreamWriter = New StreamWriter(saveFileDialog1.FileName)
                sw.Write(txtResultsGB.Text)
                sw.Close()
            End If
            saveFileDialog1.InitialDirectory = "C:\Documents and Settings\Administrator\My
Documents"
            'saveFileDialog1.AddExtension = True
        End If
    End Sub

    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ExitToolStripMenuItem.Click
        Me.Close()
    End Sub

    Private Sub msRunSB_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
msRunSB.Click
        RunSB()
    End Sub

    'Method used to enable and disable run commnads for slab and girder bridge
    Private Sub msRun_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
msRun.Click
        If tbcMain.SelectedTab.Equals(tbSlab) Then
            msRunSB.Enabled = True
            msRunGB.Enabled = False
        ElseIf tbcMain.SelectedTab.Equals(tbGirder) Then
            msRunSB.Enabled = False
            msRunGB.Enabled = True
        End If
    End Sub

    'Girder Bridge Strength Evaluation
    'Variable Declaration
    'General Information
    Dim strBridgeIdentificationNumberGB As String
    Dim strBridgeNameGB As String
    Dim strRiverNameGB As String

    'Material type
    Dim strRailingGB As String 'Railing material type
    Dim strPostGB As String 'Post material type

    'Dimensions
```

```
Dim dblLtGB As Double 'Total length of superstructure
Dim dblSGB As Double 'Span length
Dim dblWtGB As Double 'Total width of superstructure
Dim dblWcGB As Double 'Clear width of superstructure
Dim intNLGB As Integer 'Number of lanes

'Girder Dimension
Dim dblDgGB As Double 'Depth of Girder
Dim dblSgGB As Double 'c/c spacing of girders
Dim dblEgGB As Double 'End of slab to center of exterior girder
Dim dblWgGB As Double 'Girder web width
Dim intNGGB As Double 'Number of girder

'Edge beam
Dim dblDeGB As Double 'Edge beam depth
Dim dblWeGB As Double 'Edge beam width
Dim dblDAeGB As Double 'Edge beam depth beneath bottom slab level

'Diaphram
Dim dblDdeGB As Double 'Exterior diaphram depth
Dim dblDdiGB As Double 'Interior diaphram depth
Dim dblWdGB As Double 'diaphram width
Dim intNdGB As Integer 'Number of diaphrams
Dim dblSdGB As Double 'Spacing between diaphrams

'Slab and wearing surface thickness
Dim dblDsGB As Double 'Slab thickness
Dim dblDSeGB As Double 'Slab thickness near edge beam
Dim dblDwsGB As Double 'Wearing surface thickness

'Post and railing
Dim dblDrGB As Double 'Railing depth
Dim dblWrGB As Double 'Railing width
Dim dblDpGB As Double 'Post depth
Dim dblWpGB As Double 'Post width
Dim dblHpGB As Double 'Post height
Dim dblSpGB As Double 'Post spacing
Dim dblNpGB As Double 'Number of posts
Dim dblFrGB As Double 'Face of railing/curb to end of slab

'Bridge Condition
Dim strTrafficConditionGB As String 'Traffic condition
Dim strExteriorGirderConditionGB As String
Dim strInteriorGirderConditionGB As String
Dim strWearingSurfaceConditionGB As String
Dim strInspectionGB As String
Dim strRedundancyGB As String
Dim strMaintenanceGB As String
Dim dblCGGB As Double 'Concrete grade
Dim dblBGB As Double ' Nuetral axis depth

'Reinforcement
'Exterior girder
'Main reinforcement
Dim dblDbeGB As Double 'Diameter of bar
Dim dblNbeGB As Double 'Number of bars
Dim dblCbeGB As Double 'Cover of reinforcement
Dim dblYfeGB As Double 'yield stress of main bar
'Shear reinforcement
Dim dblDbseGB As Double 'diameter of bar
Dim dblSbseGB As Double 'Spacing of shear reinforcement
Dim dblYseGB As Double 'Yield stress of shear reinforcement

'Ixterior girder
'Main reinforcement
```

```
    Dim dblDbiGB As Double 'Diameter of bar
    Dim dblNbiGB As Double 'Number of bars
    Dim dblCbiGB As Double 'Cover of reinforcement
    Dim dblYfiGB As Double 'yield stress of main bar
    'Shear reinforcement
    Dim dblDbsiGB As Double 'diameter of bar
    Dim dblSbsiGB As Double 'Spacing of shear reinforcement
    Dim dblYsiGB As Double 'Yield stress of shear reinforcement

    Private Sub txtBridgeIdentificationNumberGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtBridgeIdentificationNumberGB.TextChanged
        strBridgeIdentificationNumberGB = txtBridgeIdentificationNumberGB.Text
    End Sub

    Private Sub txtBridgeNameGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtBridgeNameGB.TextChanged
        strBridgeNameGB = txtBridgeNameGB.Text
    End Sub

    Private Sub txtRiverNameGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtRiverNameGB.TextChanged
        strRiverNameGB = txtRiverNameGB.Text
    End Sub

    Private Sub cmbRailingMaterialTypeGB_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmbRailingMaterialTypeGB.SelectedIndexChanged
        strRailingGB = cmbRailingMaterialTypeGB.Text
    End Sub

    Private Sub cmbPostMaterialTypeGB_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmbPostMaterialTypeGB.SelectedIndexChanged
        strPostGB = cmbPostMaterialTypeGB.Text
    End Sub

    Private Sub txtTotalLengthOfSuperstructureGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTotalLengthOfSuperstructureGB.TextChanged
        Try
            dblLtGB = txtTotalLengthOfSuperstructureGB.Text
        Catch ex As Exception
            If txtTotalLengthOfSuperstructureGB.Text = String.Empty Or
txtTotalLengthOfSuperstructureGB.Text = "." Then
                dblLtGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Total Length of Superstructure'",
"Data Error")
                txtTotalLengthOfSuperstructureGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtSpanLengthGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtSpanLengthGB.TextChanged
        Try
            dblSGB = txtSpanLengthGB.Text
        Catch ex As Exception
            If txtSpanLengthGB.Text = String.Empty Or txtSpanLengthGB.Text = "." Then
                dblSGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Span Length'", "Data Error")
                txtSpanLengthGB.Text = String.Empty
            End If
        End Try
    End Sub
```

```
    Private Sub txtTotalWiddthOfSuperstructureGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTotalWiddthOfSuperstructureGB.TextChanged
        Try
            dblWtGB = txtTotalWiddthOfSuperstructureGB.Text
        Catch ex As Exception
            If txtTotalWiddthOfSuperstructureGB.Text = String.Empty Or
txtTotalWiddthOfSuperstructureGB.Text = "." Then
                dblWtGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Total Width of Superstructure'",
"Data Error")
                txtTotalWiddthOfSuperstructureGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtClearWidthOfSuperstructureGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtClearWidthOfSuperstructureGB.TextChanged
        Try
            dblWcGB = txtClearWidthOfSuperstructureGB.Text
        Catch ex As Exception
            If txtClearWidthOfSuperstructureGB.Text = String.Empty Or
txtClearWidthOfSuperstructureGB.Text = "." Then
                dblWcGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Clear Width of Superstructure'",
"Data Error")
                txtClearWidthOfSuperstructureGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtNumberOfLanesGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtNumberOfLanesGB.TextChanged
        Try
            intNLGB = txtNumberOfLanesGB.Text
            If txtNumberOfLanesGB.Text.Contains(".") Then
                MessageBox.Show("Please give integer values only", "Data Error")
                txtNumberOfLanesGB.Text = String.Empty
            End If
        Catch ex As Exception
            If txtNumberOfLanesGB.Text = String.Empty Then
                intNLGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Number of Lanes'", "Data Error")
                txtNumberOfLanesGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtGirderDepthGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtGirderDepthGB.TextChanged
        Try
            dblDgGB = txtGirderDepthGB.Text
        Catch ex As Exception
            If txtGirderDepthGB.Text = String.Empty Or txtGirderDepthGB.Text = "." Then
                dblDgGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Depth of Girder'", "Data Error")
                txtGirderDepthGB.Text = String.Empty
            End If
        End Try
    End Sub
```

```
    Private Sub txtGirderSpacingGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtGirderSpacingGB.TextChanged
        Try
            dblSgGB = txtGirderSpacingGB.Text
        Catch ex As Exception
            If txtGirderSpacingGB.Text = String.Empty Or txtGirderSpacingGB.Text = "." Then
                dblSgGB = 0
            Else
                MessageBox.Show("Please give correct value for 'c/c Spacing of Girder'", "Data
Error")
                txtGirderSpacingGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtEndOfSlabToCenterOfExteriorGirderGB_TextChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles txtEndOfSlabToCenterOfExteriorGirderGB.TextChanged
        Try
            dblEgGB = txtEndOfSlabToCenterOfExteriorGirderGB.Text
        Catch ex As Exception
            If txtEndOfSlabToCenterOfExteriorGirderGB.Text = String.Empty Or
txtEndOfSlabToCenterOfExteriorGirderGB.Text = "." Then
                dblEgGB = 0
            Else
                MessageBox.Show("Please give correct value for 'End of Slab to Center of Exterior
Girder'", "Data Error")
                txtEndOfSlabToCenterOfExteriorGirderGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtGirderWebWidthGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtGirderWebWidthGB.TextChanged
        Try
            dblWgGB = txtGirderWebWidthGB.Text
        Catch ex As Exception
            If txtGirderWebWidthGB.Text = String.Empty Or txtGirderWebWidthGB.Text = "." Then
                dblWgGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Girder Web Width'", "Data Error")
                txtGirderWebWidthGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtNumberOfGirdersGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtNumberOfGirdersGB.TextChanged
        Try
            intNGGB = txtNumberOfGirdersGB.Text
            If txtNumberOfGirdersGB.Text.Contains(".") Then
                MessageBox.Show("Please give integer values only", "Data Error")
                txtNumberOfGirdersGB.Text = String.Empty
            End If
        Catch ex As Exception
            If txtNumberOfGirdersGB.Text = String.Empty Then
                intNGGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Number of Girders'", "Data Error")
                txtNumberOfGirdersGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtEdgeBeamWidthGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtEdgeBeamWidthGB.TextChanged
```

```
        Try
            dblWeGB = txtEdgeBeamWidthGB.Text
        Catch ex As Exception
            If txtEdgeBeamWidthGB.Text = String.Empty Or txtEdgeBeamWidthGB.Text = "." Then
                dblWeGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Edge Beam Width'", "Data Error")
                txtEdgeBeamWidthGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtEdgeBeamDepthGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtEdgeBeamDepthGB.TextChanged
        Try
            dblDeGB = txtEdgeBeamDepthGB.Text
        Catch ex As Exception
            If txtEdgeBeamDepthGB.Text = String.Empty Or txtEdgeBeamDepthGB.Text = "." Then
                dblDeGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Edge Beam Depth'", "Data Error")
                txtEdgeBeamDepthGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtEdgeBeamDepthBeneathTheBottomSlabLevelGB_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles txtEdgeBeamDepthBeneathTheBottomSlabLevelGB.TextChanged
        Try
            dblDAeGB = txtEdgeBeamDepthBeneathTheBottomSlabLevelGB.Text
        Catch ex As Exception
            If txtEdgeBeamDepthBeneathTheBottomSlabLevelGB.Text = String.Empty Or
txtEdgeBeamDepthBeneathTheBottomSlabLevelGB.Text = "." Then
                dblDAeGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Edge Beam Depth Beneath the Bottom
Slab Level'", "Data Error")
                txtEdgeBeamDepthBeneathTheBottomSlabLevelGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtExteriorDiaphramDepthGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtExteriorDiaphramDepthGB.TextChanged
        Try
            dblDdeGB = txtExteriorDiaphramDepthGB.Text
        Catch ex As Exception
            If txtExteriorDiaphramDepthGB.Text = String.Empty Or txtExteriorDiaphramDepthGB.Text =
"." Then
                dblDdeGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Exterior Diaphram Depth'", "Data
Error")
                txtExteriorDiaphramDepthGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtInteriorDiaphramDepthGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtInteriorDiaphramDepthGB.TextChanged
        Try
            dblDdiGB = txtInteriorDiaphramDepthGB.Text
        Catch ex As Exception
            If txtInteriorDiaphramDepthGB.Text = String.Empty Or txtInteriorDiaphramDepthGB.Text =
"." Then
```

```
                dblDdiGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Interior Diaphram Depth'", "Data
Error")
                txtInteriorDiaphramDepthGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtWidthOfDiaphramGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtWidthOfDiaphramGB.TextChanged
        Try
            dblWdGB = txtWidthOfDiaphramGB.Text
        Catch ex As Exception
            If txtWidthOfDiaphramGB.Text = String.Empty Or txtWidthOfDiaphramGB.Text = "." Then
                dblWdGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Width of Diaphram'", "Data Error")
                txtWidthOfDiaphramGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtDistanceBetweenDiaphramsGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtDistanceBetweenDiaphramsGB.TextChanged
        Try
            dblSdGB = txtDistanceBetweenDiaphramsGB.Text
        Catch ex As Exception
            If txtDistanceBetweenDiaphramsGB.Text = String.Empty Or
txtDistanceBetweenDiaphramsGB.Text = "." Then
                dblSdGB = 0
            Else
                MessageBox.Show("Please give correct value for 'c/c Distance Between Diaphram'",
"Data Error")
                txtDistanceBetweenDiaphramsGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtTotalNumberOfDiaphramsGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTotalNumberOfDiaphramsGB.TextChanged
        Try
            intNdGB = txtTotalNumberOfDiaphramsGB.Text
            If txtTotalNumberOfDiaphramsGB.Text.Contains(".") Then
                MessageBox.Show("Please give integer values only", "Data Error")
                txtTotalNumberOfDiaphramsGB.Text = String.Empty
            End If
        Catch ex As Exception
            If txtTotalNumberOfDiaphramsGB.Text = String.Empty Then
                intNdGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Number of Diaphrams'", "Data
Error")
                txtTotalNumberOfDiaphramsGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtTopSlabThicknessGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTopSlabThicknessGB.TextChanged
        Try
            dblDsGB = txtTopSlabThicknessGB.Text
        Catch ex As Exception
            If txtTopSlabThicknessGB.Text = String.Empty Or txtTopSlabThicknessGB.Text = "." Then
                dblDsGB = 0
```

```
            Else
                MessageBox.Show("Please give correct value for 'Top Slab Thickness'", "Data
Error")
                txtTopSlabThicknessGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtOverhangSlabThicknessNearEdgeBeamGB_TextChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles txtOverhangSlabThicknessNearEdgeBeamGB.TextChanged
        Try
            dblDSeGB = txtOverhangSlabThicknessNearEdgeBeamGB.Text
        Catch ex As Exception
            If txtOverhangSlabThicknessNearEdgeBeamGB.Text = String.Empty Or
txtOverhangSlabThicknessNearEdgeBeamGB.Text = "." Then
                dblDSeGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Overhang Slab Thickness Near Edge
Beam'", "Data Error")
                txtOverhangSlabThicknessNearEdgeBeamGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtWearingSurfaceThicknessGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtWearingSurfaceThicknessGB.TextChanged
        Try
            If rdbNominalThicknessGB.Checked = True Then
                dblDwsGB = 1.2 * txtWearingSurfaceThicknessGB.Text
            ElseIf rdbMeasuredThicknessGB.Checked = True Then
                dblDwsGB = txtWearingSurfaceThicknessGB.Text
            End If
        Catch ex As Exception
            If txtWearingSurfaceThicknessGB.Text = String.Empty Or
txtWearingSurfaceThicknessGB.Text = "." Then
                dblDwsGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Wearing Surface Thickness'", "Data
Error")
                txtWearingSurfaceThicknessGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub rdbNominalThicknessGB_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles rdbNominalThicknessGB.CheckedChanged
        Try
            If rdbNominalThicknessGB.Checked = True Then
                dblDwsGB = 1.2 * txtWearingSurfaceThicknessGB.Text
            Else
                dblDwsGB = txtWearingSurfaceThicknessGB.Text
            End If
        Catch ex As Exception
        End Try
    End Sub

    Private Sub rdbMeasuredThicknessGB_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles rdbMeasuredThicknessGB.CheckedChanged
        Try
            If rdbMeasuredThicknessGB.Checked = False Then
                dblDwsGB = 1.2 * txtWearingSurfaceThicknessGB.Text
            Else
                dblDwsGB = txtWearingSurfaceThicknessGB.Text
            End If
        Catch ex As Exception
```

```
        End Try
    End Sub
    'Railing depth - dblDr
    Private Sub txtRailingDepthGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtRailingDepthGB.TextChanged
        Try
            dblDrGB = txtRailingDepthGB.Text
        Catch ex As Exception
            If txtRailingDepthGB.Text = String.Empty Or txtRailingDepthGB.Text = "." Then
                dblDrGB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Railing Depth'", "Data Error")
                txtRailingDepthGB.Text = String.Empty
            End If
        End Try
    End Sub

    'Railing width - dblWr
    Private Sub txtRailingWidthGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtRailingWidthGB.TextChanged
        Try
            dblWrGB = txtRailingWidthGB.Text
        Catch ex As Exception
            If txtRailingWidthGB.Text = String.Empty Or txtRailingWidthGB.Text = "." Then
                dblWrGB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Railing Width'", "Data Error")
                txtRailingWidthGB.Text = String.Empty
            End If
        End Try
    End Sub

    'Post Depth - dblDp
    Private Sub txtPostDepthGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPostDepthGB.TextChanged
        Try
            dblDpGB = txtPostDepthGB.Text
        Catch ex As Exception
            If txtPostDepthGB.Text = String.Empty Or txtPostDepthGB.Text = "." Then
                dblDpGB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Post Depth'", "Data Error")
                txtPostDepthGB.Text = String.Empty
            End If
        End Try
    End Sub

    'Post Width - dblWp
    Private Sub txtPostWidthGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPostWidthGB.TextChanged
        Try
            dblWpGB = txtPostWidthGB.Text
        Catch ex As Exception
            If txtPostWidthGB.Text = String.Empty Or txtPostWidthGB.Text = "." Then
                dblWpGB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Post Width'", "Data Error")
                txtPostWidthGB.Text = String.Empty
            End If
        End Try
    End Sub

    'Post height - dblHp
    Private Sub txtPostHeightGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPostHeightGB.TextChanged
```

```
        Try
            dblHpGB = txtPostHeightGB.Text
        Catch ex As Exception
            If txtPostHeightGB.Text = String.Empty Or txtPostHeightGB.Text = "." Then
                dblHpGB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Post Height'", "Data Error")
                txtPostHeightGB.Text = String.Empty
            End If
        End Try
    End Sub

    'Post spacing - dblSp
    Private Sub txtPostSpacingGB_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPostSpacingGB.TextChanged
        Try
            dblSpGB = txtPostSpacingGB.Text
        Catch ex As Exception
            If txtPostSpacingGB.Text = String.Empty Or txtPostSpacingGB.Text = "." Then
                dblSpGB = 0
            Else
                MessageBox.Show("Please give correct Value for 'Post Spacing'", "Data Error")
                txtPostSpacingGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtFaceOfRailingOrCurbToEndOfSlab_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtFaceOfRailingOrCurbToEndOfSlab.TextChanged
        Try
            dblFrGB = txtFaceOfRailingOrCurbToEndOfSlab.Text
        Catch ex As Exception
            If txtFaceOfRailingOrCurbToEndOfSlab.Text = String.Empty Or
txtFaceOfRailingOrCurbToEndOfSlab.Text = "." Then
                dblFrGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Total Length of Superstructure'",
"Data Error")
                txtFaceOfRailingOrCurbToEndOfSlab.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub lstTrafficConditionGB_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles lstTrafficConditionGB.SelectedIndexChanged
        strTrafficConditionGB = lstTrafficConditionGB.SelectedItem
    End Sub

    Private Sub cmbExteriorGirderConditionGB_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles cmbExteriorGirderConditionGB.SelectedIndexChanged
        strExteriorGirderConditionGB = cmbExteriorGirderConditionGB.Text
    End Sub

    Private Sub cmbInteriorGirderConditionGB_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles cmbInteriorGirderConditionGB.SelectedIndexChanged
        strInteriorGirderConditionGB = cmbInteriorGirderConditionGB.Text
    End Sub

    Private Sub cmbWearingSurfaceConditionGB_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles cmbWearingSurfaceConditionGB.SelectedIndexChanged
        strWearingSurfaceConditionGB = cmbWearingSurfaceConditionGB.Text
    End Sub

    Private Sub cmbInspectionGB_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmbInspectionGB.SelectedIndexChanged
```

```
        strInspectionGB = cmbInspectionGB.Text
    End Sub

    Private Sub cmbRedundancyGB_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmbRedundancyGB.SelectedIndexChanged
        strRedundancyGB = cmbRedundancyGB.Text
    End Sub

    Private Sub cmbMaintenanceGB_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmbMaintenanceGB.SelectedIndexChanged
        strMaintenanceGB = cmbMaintenanceGB.Text
    End Sub

    Private Sub txtConcreteGradeGB_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtConcreteGradeGB.TextChanged
        Try
            dblCGGB = txtConcreteGradeGB.Text
        Catch ex As Exception
            If txtConcreteGradeGB.Text = String.Empty Or txtConcreteGradeGB.Text = "." Then
                dblCGGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Cube Strength of Concrete'", "Data
Error")
                txtConcreteGradeGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtDiameterOfExteriorMainReinforcementGB_TextChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles txtDiameterOfExteriorMainReinforcementGB.TextChanged
        Try
            dblDbeGB = txtDiameterOfExteriorMainReinforcementGB.Text
        Catch ex As Exception
            If txtDiameterOfExteriorMainReinforcementGB.Text = String.Empty Or
txtDiameterOfExteriorMainReinforcementGB.Text = "." Then
                dblDbeGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Diameter of Bar'", "Data Error")
                txtDiameterOfExteriorMainReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtNumberOfBarsExteriorMainReinforcementGB_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles txtNumberOfBarsExteriorMainReinforcementGB.TextChanged
        Try
            dblNbeGB = txtNumberOfBarsExteriorMainReinforcementGB.Text
        Catch ex As Exception
            If txtNumberOfBarsExteriorMainReinforcementGB.Text = String.Empty Or
txtNumberOfBarsExteriorMainReinforcementGB.Text = "." Then
                dblNbeGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Number of Bar'", "Data Error")
                txtNumberOfBarsExteriorMainReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtCoverOfExteriorMainReinforcementGB_TextChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles txtCoverOfExteriorMainReinforcementGB.TextChanged
        Try
            dblCbeGB = txtCoverOfExteriorMainReinforcementGB.Text
        Catch ex As Exception
            If txtCoverOfExteriorMainReinforcementGB.Text = String.Empty Or
txtCoverOfExteriorMainReinforcementGB.Text = "." Then
```

```
                dblCbeGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Cover of Bar'", "Data Error")
                txtCoverOfExteriorMainReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtYieldStressOfExteriorMainReinforcementGB_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles txtYieldStressOfExteriorMainReinforcementGB.TextChanged
        Try
            dblYfeGB = txtYieldStressOfExteriorMainReinforcementGB.Text
        Catch ex As Exception
            If txtYieldStressOfExteriorMainReinforcementGB.Text = String.Empty Or
txtYieldStressOfExteriorMainReinforcementGB.Text = "." Then
                dblYfeGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Yield Stress of Bar'", "Data
Error")
                txtYieldStressOfExteriorMainReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtDiameterOfExteriorShearReinforcementGB_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles txtDiameterOfExteriorShearReinforcementGB.TextChanged
        Try
            dblDbseGB = txtDiameterOfExteriorShearReinforcementGB.Text
        Catch ex As Exception
            If txtDiameterOfExteriorShearReinforcementGB.Text = String.Empty Or
txtDiameterOfExteriorShearReinforcementGB.Text = "." Then
                dblDbseGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Diameter of Bar'", "Data Error")
                txtDiameterOfExteriorShearReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtSpacingOfExteriorShearReinforcementGB_TextChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles txtSpacingOfExteriorShearReinforcementGB.TextChanged
        Try
            dblSbseGB = txtSpacingOfExteriorShearReinforcementGB.Text
        Catch ex As Exception
            If txtSpacingOfExteriorShearReinforcementGB.Text = String.Empty Or
txtSpacingOfExteriorShearReinforcementGB.Text = "." Then
                dblSbseGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Spacing of Bar'", "Data Error")
                txtSpacingOfExteriorShearReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtYieldStressOfExteriorShearReinforcementGB_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles txtYieldStressOfExteriorShearReinforcementGB.TextChanged
        Try
            dblYseGB = txtYieldStressOfExteriorShearReinforcementGB.Text
        Catch ex As Exception
            If txtYieldStressOfExteriorShearReinforcementGB.Text = String.Empty Or
txtYieldStressOfExteriorShearReinforcementGB.Text = "." Then
                dblYseGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Diameter of Bar'", "Data Error")
                txtYieldStressOfExteriorShearReinforcementGB.Text = String.Empty
```

```
            End If
        End Try
    End Sub

    Private Sub txtDiameterOfInteriorMainReinforcementGB_TextChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles txtDiameterOfInteriorMainReinforcementGB.TextChanged
        Try
            dblDbiGB = txtDiameterOfInteriorMainReinforcementGB.Text
        Catch ex As Exception
            If txtDiameterOfInteriorMainReinforcementGB.Text = String.Empty Or
txtDiameterOfInteriorMainReinforcementGB.Text = "." Then
                dblDbiGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Diameter of Bar'", "Data Error")
                txtDiameterOfInteriorMainReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtNumberOfBarsInteriorMainReinforcementGB_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles txtNumberOfBarsInteriorMainReinforcementGB.TextChanged
        Try
            dblNbiGB = txtNumberOfBarsInteriorMainReinforcementGB.Text
        Catch ex As Exception
            If txtNumberOfBarsInteriorMainReinforcementGB.Text = String.Empty Or
txtNumberOfBarsInteriorMainReinforcementGB.Text = "." Then
                dblNbiGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Number of Bar'", "Data Error")
                txtNumberOfBarsInteriorMainReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtCoverOfInteriorMainReinforcementGB_TextChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles txtCoverOfInteriorMainReinforcementGB.TextChanged
        Try
            dblCbiGB = txtCoverOfInteriorMainReinforcementGB.Text
        Catch ex As Exception
            If txtCoverOfInteriorMainReinforcementGB.Text = String.Empty Or
txtCoverOfInteriorMainReinforcementGB.Text = "." Then
                dblCbiGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Cover of Bar'", "Data Error")
                txtCoverOfInteriorMainReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtYieldStressOfInteriorMainReinforcementGB_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles txtYieldStressOfInteriorMainReinforcementGB.TextChanged
        Try
            dblYfiGB = txtYieldStressOfInteriorMainReinforcementGB.Text
        Catch ex As Exception
            If txtYieldStressOfInteriorMainReinforcementGB.Text = String.Empty Or
txtYieldStressOfInteriorMainReinforcementGB.Text = "." Then
                dblYfiGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Yield Stress of Bar'", "Data
Error")
                txtYieldStressOfInteriorMainReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub
```

```
    Private Sub txtDiameterOfInteriorShearReinforcementGB_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles txtDiameterOfInteriorShearReinforcementGB.TextChanged
        Try
            dblDbsiGB = txtDiameterOfInteriorShearReinforcementGB.Text
        Catch ex As Exception
            If txtDiameterOfInteriorShearReinforcementGB.Text = String.Empty Or
txtDiameterOfInteriorShearReinforcementGB.Text = "." Then
                dblDbsiGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Diameter of Bar'", "Data Error")
                txtDiameterOfInteriorShearReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtSpacingOfInteriorShearReinforcementGB_TextChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles txtSpacingOfInteriorShearReinforcementGB.TextChanged
        Try
            dblSbsiGB = txtSpacingOfInteriorShearReinforcementGB.Text
        Catch ex As Exception
            If txtSpacingOfInteriorShearReinforcementGB.Text = String.Empty Or
txtSpacingOfInteriorShearReinforcementGB.Text = "." Then
                dblSbsiGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Spacing of Bar'", "Data Error")
                txtSpacingOfInteriorShearReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    Private Sub txtYieldStressOfInteriorShearReinforcementGB_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles txtYieldStressOfInteriorShearReinforcementGB.TextChanged
        Try
            dblYsiGB = txtYieldStressOfInteriorShearReinforcementGB.Text
        Catch ex As Exception
            If txtYieldStressOfInteriorShearReinforcementGB.Text = String.Empty Or
txtYieldStressOfInteriorShearReinforcementGB.Text = "." Then
                dblYsiGB = 0
            Else
                MessageBox.Show("Please give correct value for 'Diameter of Bar'", "Data Error")
                txtYieldStressOfInteriorShearReinforcementGB.Text = String.Empty
            End If
        End Try
    End Sub

    'Display input values
    Public Sub DisplayInputValuesGB()
        txtResultsGB.Text = "GIRDER BRIDGE STRENGTH EVALUATION"
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "INPUT VALUES")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "GENERAL INFORMATION")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Bridge Identification Number:    " &
strBridgeIdentificationNumberGB)
        txtResultsGB.AppendText(Environment.NewLine & "Bridge Name:    " & strBridgeNameGB)
        txtResultsGB.AppendText(Environment.NewLine & "River Name:    " & strRiverNameGB)
        txtResultsGB.AppendText(Environment.NewLine & "Railing Material Type:   " & strRailingGB)
        txtResultsGB.AppendText(Environment.NewLine & "Post Material Type:  " & strPostGB)
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "DIMENSIONS")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Total Length of Superstructure   =   " &
dblLtGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Span Length  =   " & dblSGB & " m")
```

```
        txtResultsGB.AppendText(Environment.NewLine & "Total Width of Superstructure   =   " &
dblWtGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Clear With of Superstructure   =   " &
dblWcGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Number of Lanes   =   " & intNLGB)
        txtResultsGB.AppendText(Environment.NewLine & "Depth of Girder   =   " & dblDgGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "c/c of Girder Spacing   =   " & dblSgGB & "
m")
        txtResultsGB.AppendText(Environment.NewLine & "End of Slab to Center of Exterior Girder:
" & dblEgGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Girder Web Width   =   " & dblWgGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Number of Girders   =   " & intNGGB)
        txtResultsGB.AppendText(Environment.NewLine & "Edge Beam Depth   =   " & dblDeGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Edge Beam Width   =   " & dblWeGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Edge Beam Depth Beneath the Bottom Slab
Level   =   " & dblDAeGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Diaphram Depth   =   " & dblDdeGB
& " m")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Diaphram Depth   =   " & dblDdiGB
& " m")
        txtResultsGB.AppendText(Environment.NewLine & "Diaphram Width   =   " & dblWdGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Number of Diaphrams   =   " & intNdGB)
        txtResultsGB.AppendText(Environment.NewLine & "c/c Distance Between Diaphrams   =   " &
dblSdGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Top Slab Thickness   =   " & dblDsGB & "
m")
        txtResultsGB.AppendText(Environment.NewLine & "Slab Thickness Near Edge Beam   =   " &
dblDSeGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Wearing Surface Thickness   =   " &
dblDwsGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Railing Depth   =   " & dblDrGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Railing Width   =   " & dblWrGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Post Depth   =   " & dblDpGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Post Width   =   " & dblWpGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Post Height   =   " & dblHpGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Post Spacing   =   " & dblSpGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "Face of railing/curb to end of slab   =   "
& dblFrGB & " m")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "BRIDGE CONDITION")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Traffic Condition:   " &
strTrafficConditionGB)
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder Condition:   " &
strExteriorGirderConditionGB)
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder Condition:   " &
strInteriorGirderConditionGB)
        txtResultsGB.AppendText(Environment.NewLine & "Wearing Surface Condition:   " &
strWearingSurfaceConditionGB)
        txtResultsGB.AppendText(Environment.NewLine & "Inspection:  " & strInspectionGB)
        txtResultsGB.AppendText(Environment.NewLine & "Redundancy:  " & strRedundancyGB)
        txtResultsGB.AppendText(Environment.NewLine & "Maintenance: " & strMaintenanceGB)
        txtResultsGB.AppendText(Environment.NewLine & "Concrete Grade   =   " & dblCGGB & " MPa")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "REINFORCEMENT")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Main Reinforcement Diameter   =
" & dblDbeGB & " mm")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Main Reinforcement Number of Bars
=   " & dblNbeGB)
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Main Reinforcement Cover   =   " &
dblCbeGB & " mm")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Main Reinforcement Yield Stress
=   " & dblYfeGB & " MPa")
```

```
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Shear Reinforcement Diameter   =
" & dblDbseGB & " mm")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Shear Reinforcement Spacing   =
" & dblSbseGB & " mm")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Shear Reinforcement Yield Stress
=   " & dblYseGB & " MPa")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Main Reinforcement Diameter   =
" & dblDbiGB & " mm")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Main Reinforcement Number of Bars
=   " & dblNbiGB)
        txtResultsGB.AppendText(Environment.NewLine & "Interior Main Reinforcement Cover   =   " &
dblCbiGB & " mm")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Main Reinforcement Yield Stress
=   " & dblYfiGB & " MPa")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Shear Reinforcement Diameter   =
" & dblDbsiGB & " mm")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Shear Reinforcement Spacing   =
" & dblSbsiGB & " mm")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Shear Reinforcement Yield Stress
=   " & dblYsiGB & " MPa")
    End Sub

    'Display calculated value
    Public Sub DisplayResultsGB()
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "OUTPUT VALUES")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Number of lanes   =   " & intNLGB)
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "UNIT WEIGTHTS")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Unit weight of Railing   =   " & dblGamaRGB
& " KN/m3")
        txtResultsGB.AppendText(Environment.NewLine & "Unit weight of Post   =   " & dblGamaPGB &
" KN/m3")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "FACTORS")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Dead Load Factor   =   " & dblgamaDLGB)
        txtResultsGB.AppendText(Environment.NewLine & "Live Load Factor   =   " & dblgamaLLGB)
        txtResultsGB.AppendText(Environment.NewLine & "Strength Reduction Factor - Interior Girder
=   " & dblPhiiGB)
        txtResultsGB.AppendText(Environment.NewLine & "Strength Reduction Factor - Exterior Girder
=   " & dblPhieGB)
        txtResultsGB.AppendText(Environment.NewLine & "Impact Factor   =   " & dblIFGB)
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "DEAD LOADS")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder Dead Load   =   " &
Math.Round(dblDLiGB, 2) & " KN/m")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder Dead Load   =   " &
Math.Round(dblDLeGB, 2) & " KN/m")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Diaphram Dead Load (Interior of
Span)   =   " & Math.Round(dblDLdiisGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Diaphram Dead Load (End of Span)
=   " & Math.Round(dblDLdiesGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Diaphram Dead Load (Interior of
Span)   =   " & Math.Round(dblDLdeisGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Diaphram Dead Load (End of Span)
=   " & Math.Round(dblDLdeesGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "DEAD LOAD SHEAR FORCE AND MOMENT RESULTS")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder - Distributed Load Shear
Force and Moment")
```

```
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "V(KN)" & "                 " & "M(KN-m)")
        For intI As Integer = 0 To 10
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblVdiGB(intI), 2) & "
" & Math.Round(dblMdiGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder - Distributed Load Shear
Force and Moment")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "V(KN)" & "                 " & "M(KN-m)")
        For intI As Integer = 0 To 10
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblVdeGB(intI), 2) & "
" & Math.Round(dblMdeGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder - Diaphram Load Shear Force
and Moment")
        txtResultsGB.AppendText(Environment.NewLine & "V(KN)" & "                 " & "M(KN-m)")
        For intI As Integer = 0 To 10
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblVddiGB(intI), 2) & "
" & Math.Round(dblMddiGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder - Diaphram Load Force Shear
and Moment")
        txtResultsGB.AppendText(Environment.NewLine & "V(KN)" & "                 " & "M(KN-m)")
        For intI As Integer = 0 To 10
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblVddeGB(intI), 2) & "
" & Math.Round(dblMddeGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "DISTRIBUTION FACTORS")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Moment Distribution Factor   =   " &
dblDFmi)
        txtResultsGB.AppendText(Environment.NewLine & "Shear Distribution Factor   =   " &
dblDFsi)
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Moment Distribution Factor   =   " &
dblDFme)
        txtResultsGB.AppendText(Environment.NewLine & "Shear Distribution Factor   =   " &
dblDFse)
        txtResultsGB.AppendText(Environment.NewLine & "")

        'Live load shear force and moment
        txtResultsGB.AppendText(Environment.NewLine & "LIVE LOAD SHEAR FORCE (KN) AND MOMENT (KN-
m)RESULTS")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Due to truck type 3-1")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblV3iGB(intI), 2) & "
" & Math.Round(dblM3iGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Due to truck type 3_2")
        txtResultsGB.AppendText(Environment.NewLine & "")
```

```
        txtResultsGB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblV3_2iGB(intI), 2) & "
" & Math.Round(dblM3_2iGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Due to truck type 3_3")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblV3_3iGB(intI), 2) & "
" & Math.Round(dblM3_3iGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Due to Legal Lane Loading")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblVLLiGB(intI), 2) & "
" & Math.Round(dblMLLiGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Due to truck type 3-1")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblV3eGB(intI), 2) & "
" & Math.Round(dblM3eGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Due to truck type 3_2")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblV3_2eGB(intI), 2) & "
" & Math.Round(dblM3_2eGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Due to truck type 3_3")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblV3_3eGB(intI), 2) & "
" & Math.Round(dblM3_3eGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Due to Legal Lane Loading")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "V (KN)" & "          " & "M (KN-m)")
        For intI As Integer = 0 To 11
            txtResultsGB.AppendText(Environment.NewLine & Math.Round(dblVLLeGB(intI), 2) & "
" & Math.Round(dblMLLeGB(intI), 2))
        Next
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "MAXIMUM SHEAR FORCE AND MOMENT FOR EACH
LOAD CASE")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-1
=   " & Math.Round(dblMax_V3iGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-1   =
" & Math.Round(dblMax_M3iGB, 2) & " KN-m")
```

```
        txtResultsGB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-2
=   " & Math.Round(dblMax_V3_2iGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-2   =
" & Math.Round(dblMax_M3_2iGB, 2) & " KN-m")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-3
=   " & Math.Round(dblMax_V3_3iGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-3 = " &
Math.Round(dblMax_M3_3iGB, 2) & " KN-m")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum shear force due to truck type legal
lane load   =   " & Math.Round(dblMax_VLLiGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum moment due to truck type legal lane
load   =   " & Math.Round(dblMax_MLLiGB, 2) & " KN-m")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-1
=   " & Math.Round(dblMax_V3eGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-1   =
" & Math.Round(dblMax_M3eGB, 2) & " KN-m")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-2
=   " & Math.Round(dblMax_V3_2eGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-2   =
" & Math.Round(dblMax_M3_2eGB, 2) & " KN-m")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum shear force due to truck type 3-3
=   " & Math.Round(dblMax_V3_3eGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum moment due to truck type 3-3   =
" & Math.Round(dblMax_M3_3eGB, 2) & " KN-m")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum shear force due to truck type legal
lane load   =   " & Math.Round(dblMax_VLLeGB, 2) & " KN")
        txtResultsGB.AppendText(Environment.NewLine & "Maximum moment due to truck type legal lane
load   =   " & Math.Round(dblMax_MLLeGB, 2) & " KN-m")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "STRENGTH CALCULATIONS")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Shear Strength")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder   =   " &
Math.Round(dblVriGB, 2) & "KN")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder   =   " &
Math.Round(dblVreGB, 2) & "KN")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Moment Strength")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder   =   " &
Math.Round(dblMriGB, 2) & " KN-m")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder   =   " &
Math.Round(dblMreGB, 2) & " KN-m")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "RATING FACTORS")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Interior Girder")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Rating factor for shear")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3-1: RF   =   " &
Math.Round(dblRFV3iGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3_2: RF   =   " &
Math.Round(dblRFV3_2iGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3_3: RF   =   " &
Math.Round(dblRFV3_3iGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Legal lane loading: RF   =   " &
Math.Round(dblRFVLLiGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Rating factor for moment")
        txtResultsGB.AppendText(Environment.NewLine & "")
```

```
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3-1: RF   =   " &
Math.Round(dblRFM3iGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3_2: RF   =   " &
Math.Round(dblRFM3_2iGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3_3: RF   =   " &
Math.Round(dblRFM3_3iGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Legal lane loading: RF   =   " &
Math.Round(dblRFMLLiGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Exterior Girder")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Rating factor for shear")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3-1: RF   =   " &
Math.Round(dblRFV3eGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3_2: RF   =   " &
Math.Round(dblRFV3_2eGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3_3: RF   =   " &
Math.Round(dblRFV3_3eGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Legal lane loading: RF   =   " &
Math.Round(dblRFVLLeGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Rating factor for moment")
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3-1: RF   =   " &
Math.Round(dblRFM3eGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3_2: RF   =   " &
Math.Round(dblRFM3_2eGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Truck Type 3_3: RF   =   " &
Math.Round(dblRFM3_3eGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "Legal lane loading: RF   =   " &
Math.Round(dblRFMLLeGB, 2))
        txtResultsGB.AppendText(Environment.NewLine & "")
        txtResultsGB.AppendText(Environment.NewLine & "RATING FACTOR   =   " & Math.Round(dblRFGB,
2))
    End Sub
    'Variable Declaration
    'Unit Wieght
    Dim dblGamaRGB As Double 'Unit weigth for railing
    Dim dblGamaPGB As Double 'Unit weight for post

    'Load, strength reduction and impact factors
    Dim dblgamaDLGB As Double = 1.2 'Dead load factor
    Dim dblgamaLLGB As Double 'Live load factor
    Dim dblPhiiGB, dblPhieGB As Double 'Strength reduction factor
    Dim dblIFGB As Double 'Impact factor

    'Dead load
    Dim dblDLiGB, dblDLeGB As Double 'Girder Distributed Dead Load
    Dim dblDLdiisGB, dblDLdiesGB, dblDLdeisGB, dblDLdeesGB As Double 'Diaphram Dead Load

    'Distribution Factors for live load
    Dim dblDFmi, dblDFsi, dblDFme, dblDFse As Double

    'Variable Declaration - Array for shear force and moment
    'Dead load - distributed
    Dim dblVdeGB(10), dblVdiGB(10), dblMdeGB(10), dblMdiGB(10) As Double
    'Diaphram Dead load
    Dim dblVddeGB(10), dblVddiGB(10), dblMddeGB(10), dblMddiGB(10) As Double
    'Middle and edge strip- truck type 3
    Dim dblV3iGB(11), dblM3iGB(11), dblV3eGB(11), dblM3eGB(11) As Double
    'Middle and edge strip - truck type 3-2
    Dim dblV3_2iGB(11), dblM3_2iGB(11), dblV3_2eGB(11), dblM3_2eGB(11) As Double
    'Middle and edge strip - truck type 3-3
    Dim dblV3_3iGB(11), dblM3_3iGB(11), dblV3_3eGB(11), dblM3_3eGB(11) As Double
```

```
    'Middle and edge strip 'LLL - legal lane loading
    Dim dblVLLiGB(11), dblMLLiGB(11), dblVLLeGB(11), dblMLLeGB(11) As Double 'Live + dead load
shear force
    'Maximum shear force and moment for each case
    Dim dblMax_V3iGB, dblMax_V3_2iGB, dblMax_V3_3iGB, dblMax_VLLiGB As Double 'Interior girder -
shear
    Dim dblMax_M3iGB, dblMax_M3_2iGB, dblMax_M3_3iGB, dblMax_MLLiGB As Double 'Interior girder -
moment
    Dim dblMax_V3eGB, dblMax_V3_2eGB, dblMax_V3_3eGB, dblMax_VLLeGB As Double 'Exterior girder -
shear
    Dim dblMax_M3eGB, dblMax_M3_2eGB, dblMax_M3_3eGB, dblMax_MLLeGB As Double 'Exterior girder -
moment

    'Maximum shear force and moment out of all cases
    Dim dblViGB, dblMiGB, dblVeGB, dblMeGB As Double 'Maximum factored shear force and moment
    'Factored shear force and moment resistance
    Dim dblVriGB, dblMriGB, dblVreGB, dblMreGB As Double
    'Factored shear and moment resistance minus dead load effect
    Dim dblVrdiGB, dblMrdiGB, dblVrdeGB, dblMrdeGB As Double
    'Effective depth of girder
    Dim dblDeiGB, dblDeeGB As Double

    'Rating Factors
    Dim dblRFV3iGB, dblRFV3_2iGB, dblRFV3_3iGB, dblRFVLLiGB, dblRFM3iGB, dblRFM3_2iGB,
dblRFM3_3iGB, dblRFMLLiGB As Double
    Dim dblRFV3eGB, dblRFV3_2eGB, dblRFV3_3eGB, dblRFVLLeGB, dblRFM3eGB, dblRFM3_2eGB,
dblRFM3_3eGB, dblRFMLLeGB As Double
    Dim dblRFGB As Double

    'DeadLoadGB - Method for Dead Load Calculation
    Public Sub DeadLoadGB(ByVal strGirder As String)
        Dim dblDLre, dblDLpe, dblDLebe, dblDLse, dblDLge, dblDLwse As Double
        Dim dblDLsi, dblDLgi, dblDLwsi As Double
        Select Case strGirder
            Case "Exterior"
                dblDLre = dblGamaRGB * dblDrGB * dblWrGB
                dblDLpe = dblGamaPGB * dblHpGB * dblDpGB * dblWpGB / dblSpGB
                dblDLebe = dblGamaC * dblDeGB * dblWeGB
                dblDLse = dblGamaC * (0.5 * (dblEgGB - 0.5 * (dblWgGB + dblWeGB)) * (dblDsGB +
dblDSeGB) + 0.5 * (dblSgGB - dblWgGB) * dblDsGB)
                dblDLge = dblGamaC * dblWgGB * dblDgGB
                dblDLwse = dblGamaWS * dblDwsGB * (0.5 * dblSgGB + dblEgGB)
                dblDLeGB = dblgamaDLGB * (dblDLre + dblDLpe + dblDLebe + dblDLse + dblDLge +
dblDLwse)
            Case "Interior"
                dblDLsi = dblGamaC * dblDsGB * dblSgGB
                dblDLgi = dblGamaC * (dblDgGB - dblDsGB) * dblWgGB
                dblDLwsi = dblGamaWS * dblDwsGB * dblSgGB
                dblDLiGB = dblgamaDLGB * (dblDLsi + dblDLgi + dblDLwsi)
        End Select
    End Sub

    'DiaphramDeadLoadGB - Method to calculate diaphram load
    Public Sub DiaphramDeadLoadGB(ByVal strGirder As String)
        Select Case strGirder
            Case "Exterior"
                dblDLdeisGB = dblgamaDLGB * dblGamaC * dblDdiGB * dblWdGB * 0.5 * dblSgGB
                dblDLdeesGB = dblgamaDLGB * dblGamaC * dblDdeGB * dblWdGB * 0.5 * dblSgGB
            Case "Interior"
                dblDLdiisGB = dblgamaDLGB * dblGamaC * dblDdiGB * dblWdGB * dblSgGB
                dblDLdiesGB = dblgamaDLGB * dblGamaC * dblDdeGB * dblWdGB * dblSgGB
        End Select
    End Sub

    'DiaphramShearAndMomentGB - method to calculate diaphram shear force and moment on girder
```

```
    Public Sub DiaphramShearAndMomentGB(ByVal dblDiaphramLoad As Double, ByVal
intNumberOfDiaphrams As Integer, ByVal strGirderLocation As String)
        Dim dblA() As Double = {0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5}
        Select Case strGirderLocation
            Case "Exterior"
                Select Case intNumberOfDiaphrams
                    Case Is < 3
                        For intI As Integer = 0 To 10
                            dblVddeGB(intI) = 0
                            dblMddeGB(intI) = 0
                        Next
                    Case 3
                        For intI As Integer = 0 To 10
                            dblVddeGB(intI) = dblDiaphramLoad / 2
                            dblMddeGB(intI) = dblDiaphramLoad * dblA(intI) * dblSGB / 2
                        Next
                    Case Is >= 4
                        For intI As Integer = 0 To 10
                            If dblA(intI) * dblSGB <= dblSGB / 3 Then
                                dblVddeGB(intI) = dblDiaphramLoad
                                dblMddeGB(intI) = dblDiaphramLoad * dblA(intI) * dblSGB
                            Else
                                dblVddeGB(intI) = 0
                                dblMddeGB(intI) = dblDiaphramLoad * dblSGB / 3
                            End If
                        Next
                End Select
            Case "Interior"
                Select Case intNumberOfDiaphrams
                    Case Is < 3
                        For intI As Integer = 0 To 10
                            dblVddiGB(intI) = 0
                            dblMddiGB(intI) = 0
                        Next
                    Case 3
                        For intI As Integer = 0 To 10
                            dblVddiGB(intI) = dblDiaphramLoad / 2
                            dblMddiGB(intI) = dblDiaphramLoad * dblA(intI) * dblSGB / 2
                        Next
                    Case Is >= 4
                        For intI As Integer = 0 To 10
                            If dblA(intI) * dblSGB <= dblSGB / 3 Then
                                dblVddiGB(intI) = dblDiaphramLoad
                                dblMddiGB(intI) = dblDiaphramLoad * dblA(intI) * dblSGB
                            Else
                                dblVddiGB(intI) = 0
                                dblMddiGB(intI) = dblDiaphramLoad * dblSGB / 3
                            End If
                        Next
                        'Case Is < 4
                End Select
        End Select
    End Sub

    'DistributionFactorGB - Method used to calculate distribution factor for interior girder
    Public Sub DistributionFactorGB()
        'Calculate Kg
        Dim dblKg, dblI, dblA, dblEg As Double
        dblI = dblDgGB ^ 3 * dblWgGB / 12
        dblA = dblDgGB * dblWgGB
        dblEg = (0.5 * dblDgGB - 0.5 * dblDsGB)
        dblKg = dblI + dblA * dblEg ^ 2

        'Interior Girder
        'One design lane loaded
```

```
        Dim dblDFmi1 As Double = 0.06 + (dblSgGB / 4.3) ^ 0.4 * (dblSgGB / dblSGB) ^ 0.3 * (dblKg
/ (dblSGB * dblDsGB ^ 3)) ^ 0.1
        Dim dblDFsi1 As Double = 0.36 + dblSgGB / 7.6
        'Lever Rule
        Dim dblDFLi1 As Double
        If dblSgGB <= 1.8 Then
            dblDFLi1 = 1.2 * 0.5
        Else
            dblDFLi1 = 1.2 * 0.5 / dblSgGB * (2 * dblSgGB - 1.8)
        End If

        'Two or more design lanes loaded
        Dim DblDFmi2 As Double = 0.075 + (dblSgGB / 2.9) ^ 0.6 * (dblSgGB / dblSGB) ^ 0.2 * (dblKg
/ (dblSGB * dblDsGB ^ 3)) ^ 0.1
        Dim dblDFsi2 As Double = 0.2 + dblSgGB / 3.6 - (dblSgGB / 10.7) ^ 2
        'Lever rule
        Dim dblDFLi2 As Double
        Dim dblWi1, dblWi2, dblWi3, dblWi4 As Double
        If dblFrGB + 0.6 <= dblEgGB Then
            dblWi1 = 0
        ElseIf dblFrGB + 0.6 > dblEg Then
            dblWi1 = dblFrGB + 0.6 - dblEgGB
        End If
        If dblFrGB + 2.4 <= dblEgGB Then
            dblWi2 = 0
        ElseIf dblFrGB + 2.4 > dblEgGB And dblFrGB + 2.4 <= dblSgGB + dblEgGB Then
            dblWi2 = dblFrGB + 2.4 - dblEgGB
        ElseIf dblFrGB + 2.4 > dblSgGB + dblEgGB Then
            dblWi2 = 2 * dblSgGB + dblEgGB - (dblFrGB + 2.4)
        End If
        If dblFrGB + 3.6 <= dblSgGB + dblEgGB Then
            dblWi3 = dblFrGB + 3.6 - dblEgGB
        ElseIf dblFrGB + 3.6 > dblSgGB + dblEgGB Then
            dblWi3 = 2 * dblSgGB + dblEgGB - (dblFrGB + 3.6)
        End If
        If dblFrGB + 5.4 < 2 * dblSgGB + dblEgGB Then
            dblWi4 = 2 * dblSgGB + dblEgGB - (dblFrGB + 5.4)
        ElseIf dblFrGB + 5.4 >= 2 * dblSgGB + dblEgGB Then
            dblWi4 = 0
        End If
        dblDFLi2 = 0.5 / dblSgGB * (dblWi1 + dblWi2 + dblWi3 + dblWi4)
        'MessageBox.Show("Lever rule DF for interior girder moment two lane= " & dblDFLi2)

        If intNLGB = 1 And intNGGB >= 4 Then
            dblDFmi = dblDFmi1
            dblDFsi = dblDFsi1
        End If
        If intNLGB >= 2 And intNGGB >= 4 Then
            dblDFmi = Math.Max(dblDFmi1, DblDFmi2)
            dblDFsi = Math.Max(dblDFsi1, dblDFsi2)
        End If
        If intNLGB = 1 And intNGGB = 3 Then
            dblDFmi = Math.Min(dblDFmi1, dblDFLi1)
            dblDFsi = dblDFLi1
        End If
        If intNLGB >= 2 And intNGGB = 3 Then
            dblDFmi = Math.Min(Math.Max(dblDFmi1, DblDFmi2), Math.Max(dblDFLi1, dblDFLi2))
            dblDFsi = Math.Max(dblDFLi1, dblDFLi2)
        End If

        'Exterior Girder
        'One lane loaded
        'Lever rule
        Dim dblWe1f, dblWe2f As Double ' f-first: that is one lane loaded
        dblWe1f = dblSgGB + dblEgGB - (dblFrGB + 0.6)
```

```
        If dblFrGB + 2.4 > dblSgGB + dblEgGB Then
            dblWe2f = 0
        Else
            dblWe2f = dblSgGB + dblEgGB - (dblFrGB + 2.4)
        End If
        Dim dblDFLe1 As Double = 1.2 * 0.5 / dblSgGB * (dblWe1f + dblWe2f)

        'Two or more lane loaded
        Dim dblDe As Double = dblEgGB - dblFrGB
        Dim dblEm As Double = 0.77 + dblDe / 2.8
        Dim dblEs As Double = 0.6 + dblDe / 3
        Dim dblDFme2 As Double = dblEm * DblDFmi2
        Dim dblDFse2 As Double = dblEs * dblDFsi2
        'Lever rule
        Dim dblWe1s, dblWe2s, dblWe3s, dblWe4s As Double 's-second that is two or more lane
        dblWe1s = dblSgGB + dblEgGB - (dblFrGB + 0.6)
        If dblFrGB + 2.4 >= dblSgGB + dblEgGB Then
            dblWe2s = 0
        Else
            dblWe2s = dblSgGB + dblEgGB - (dblFrGB + 2.4)
        End If
        If dblFrGB + 3.6 >= dblSgGB + dblEgGB Then
            dblWe3s = 0
        Else
            dblWe3s = dblSgGB + dblEgGB - (dblFrGB + 3.6)
        End If
        If dblFrGB + 5.4 >= dblSgGB + dblEgGB Then
            dblWe4s = 0
        Else
            dblWe4s = dblSgGB + dblEgGB - (dblFrGB + 5.4)
        End If
        Dim dblDFLe2 = 0.5 / dblSgGB * (dblWe1s + dblWe2s + dblWe3s + dblWe4s)

        If intNLGB = 1 Then
            dblDFme = dblDFLe1
            dblDFse = dblDFLe1
        End If
        If intNLGB >= 2 And intNGGB >= 4 Then
            dblDFme = Math.Max(dblDFLe1, dblDFme2)
            dblDFse = Math.Max(dblDFLe1, dblDFse2)
        End If
        If intNLGB >= 2 And intNGGB = 3 Then
            dblDFme = Math.Min(dblDFme2, Math.Max(dblDFLe1, dblDFLe2))
            dblDFse = Math.Max(dblDFLe1, dblDFLe2)
        End If
    End Sub

    'ShearStrengthGB - method for Shear Strength Calculation
    Public Sub ShearStrengthGB(ByVal InteriorOrExterior As String)
        Dim dblNbPWg, dblSbh, dblSbv, dblDb, dblDbs, dblNb, dblSbs, dblCb, dblYf, dblYs, dblDGB As
Double
        Select Case InteriorOrExterior
            Case "Interior"
                dblDb = dblDbiGB : dblDbs = dblDbsiGB : dblNb = dblNbiGB : dblSbs = dblSbsiGB :
dblCb = dblCbiGB : dblYf = dblYfiGB : dblYs = dblYsiGB
            Case "Exterior"
                dblDb = dblDbeGB : dblDbs = dblDbseGB : dblNb = dblNbeGB : dblSbs = dblSbseGB :
dblCb = dblCbeGB : dblYf = dblYfeGB : dblYs = dblYseGB
        End Select
        'Determination for effective depth d(dblD)
        dblSbh = Math.Max(1.5 * dblDb, 38)
        dblSbv = Math.Max(dblDb, 25)
        dblNbPWg = (dblWgGB * 1000.0 - 100 + dblSbh) / (dblDb + dblSbh)
        If dblNb <= dblNbPWg Then
            dblDGB = (dblDgGB * 1000.0 - dblCb - dblDb / 2) / 1000.0
```

```
        ElseIf dblNb > dblNbPWg And dblNb <= 2 * dblNbPWg Then
            dblDGB = (dblDgGB * 1000.0 - dblCb - dblDb - dblSbv / 2) / 1000.0
        ElseIf dblNb > 2 * dblNbPWg And dblNb <= 3 * dblNbPWg Then
            dblDGB = (dblDgGB * 1000.0 - dblCb - dblDb - dblSbv - dblDb / 2) / 1000.0
        ElseIf dblNb > 3 * dblNbPWg And dblNb <= 4 * dblNbPWg Then
            dblDGB = (dblDgGB * 1000.0 - dblCb - dblDb - dblSbv - dblDb - dblSbv / 2) / 1000.0
        ElseIf dblNb > 4 * dblNbPWg And dblNb <= 5 * dblNbPWg Then
            dblDGB = (dblDgGB * 1000.0 - dblCb - dblDb - dblSbv - dblDb - dblSbv - dblDb / 2) /
1000.0
        ElseIf dblNb > 5 * dblNbPWg And dblNb <= 6 * dblNbPWg Then
            dblDGB = (dblDgGB * 1000.0 - dblCb - dblDb - dblSbv - dblDb - dblSbv - dblDb - dblSbv
/ 2) / 1000.0
        End If

        Dim dblVrc, dblVrs, dblVr As Double
        dblVrc = 0.17 * Math.Sqrt(0.8 * dblCGGB) * dblWgGB * dblDGB * 1000.0
        Dim dblAv As Double = Math.PI * dblDbs ^ 2 / 4 * 2
        dblVrs = dblAv * dblYs * dblDGB / dblSbs
        dblVr = dblVrc + dblVrs
        Select Case InteriorOrExterior
            Case "Interior"
                dblVriGB = dblPhiiGB * dblVr
                dblDeiGB = dblDGB
            Case "Exterior"
                dblVreGB = dblPhieGB * dblVr
                dblDeeGB = dblDGB
        End Select
    End Sub
    'MomentStrengthGB - method use to calculate moment capacity for girder bridge
    Public Sub MomentStrengthGB(ByVal InteriorOrExterior As String)
        If 0.8 * dblCGGB <= 28 Then : dblBGB = 0.85
        Else : dblBGB = 0.85 - 0.05 / 7 * (0.8 * dblCGGB - 28) : End If
        If dblBGB < 0.65 Then dblBGB = 0.65
        Dim dblAb, dblX, dblYf, dblMn, dblDGB, dblBeffi, dblBeffe, dblRHOb As Double
        dblBeffi = Math.Min(1 / 4 * dblSGB, Math.Min(12 * dblDsGB + dblWgGB / 2, dblSgGB))
        Select Case InteriorOrExterior
            Case "Interior"
                dblDGB = dblDeiGB : dblYf = dblYfiGB
                dblRHOb = 0.75 * 0.7225 * (600 / (600 + dblYf)) * 0.8 * dblCGGB / dblYf
                dblAb = Math.Min(0.25 * Math.PI * dblDbiGB ^ 2 * dblNbiGB, dblRHOb * dblDGB *
dblBeffi * 1000000.0)
                dblX = dblAb * dblYf / (dblBGB * dblBeffi * 0.85 * 0.8 * dblCGGB) * 0.000001
                If dblBGB * dblX <= dblDsGB Then
                    dblMn = dblAb * dblYf * (dblDGB - dblBGB * dblX / 2) * 0.001
                Else
                    dblMn = (0.85 * 0.8 * dblCGGB * (dblBeffi - dblWgGB) * dblDsGB * (dblDGB -
dblBGB * dblX / 2) + 0.85 * 0.8 * dblCGGB * dblWgGB * dblBGB * dblX * (dblDGB - dblBGB * dblX /
2)) * 1000
                End If
                dblMriGB = dblPhiiGB * dblMn
            Case "Exterior"
                dblDGB = dblDeeGB : dblYf = dblYfiGB
                dblRHOb = 0.75 * 0.7225 * (600 / (600 + dblYf)) * 0.8 * dblCGGB / dblYf
                dblBeffe = 0.5 * dblBeffi + Math.Min(1 / 8 * dblSGB, Math.Min(6 * dblDsGB +
dblWgGB / 2, (dblEgGB - dblWgGB / 2)))
                dblAb = Math.Min(0.25 * Math.PI * dblDbeGB ^ 2 * dblNbeGB, dblRHOb * dblDGB *
dblBeffe * 1000000.0)
                dblX = dblAb * dblYf / (dblBGB * dblBeffe * 0.85 * 0.8 * dblCGGB) * 0.000001
                If dblBGB * dblX <= dblDsGB Then
                    dblMn = dblAb * dblYf * (dblDGB - dblBGB * dblX / 2) * 0.001
                Else
                    dblMn = (0.85 * 0.8 * dblCGGB * (dblBeffe - dblWgGB) * dblDsGB * (dblDGB -
dblBGB * dblX / 2) + 0.85 * 0.8 * dblCGGB * dblWgGB * dblBGB * dblX * (dblDGB - dblBGB * dblX /
2)) * 1000
                End If
```

```
                dblMreGB = dblPhieGB * dblMn
                'MessageBox.Show("rhob = " & dblRHOb & " " & "Beffe = " & dblBeffe & " " & "Ab = "
& dblAb)
        End Select
    End Sub
    'Run the anaylsis for girder bridge
    Public Sub RunGB()
        tbcGirder.SelectedTab = tbResultsGB
        txtResultsGB.Clear()
        DisplayInputValuesGB()

        NumberOfLanes("Girder")
        'Unit wieght
        UnitWeight(strRailingGB, strPostGB, "Girder")

        'Live load and strength reduction Impact factor
        LiveLoadFactor(strTrafficConditionGB, "Girder")
        StrengthReductionFactor(strRedundancyGB, strExteriorGirderConditionGB, strInspectionGB,
strMaintenanceGB, "Girder", "Exterior")
        StrengthReductionFactor(strRedundancyGB, strInteriorGirderConditionGB, strInspectionGB,
strMaintenanceGB, "Girder", "Interior")
        ImpactFactor(strWearingSurfaceConditionGB, "Girder")

        'Distributed Dead load calculation
        DeadLoadGB("Exterior")
        DeadLoadGB("Interior")
        'Diaphram Dead Load Calculation
        DiaphramDeadLoadGB("Exterior")
        DiaphramDeadLoadGB("Interior")

        'Dead Load shear force and moment
        DeadLoadShearAndMoment(dblDLeGB, "Girder", "Exterior")
        DeadLoadShearAndMoment(dblDLiGB, "Girder", "Interior")
        'Diaph ram dead load shear force and moment
        DiaphramShearAndMomentGB(dblDLdeisGB, intNdGB, "Exterior")
        DiaphramShearAndMomentGB(dblDLdiisGB, intNdGB, "Interior")
        'Calculate distribution factors
        DistributionFactorGB()

        'Live load anlysis - Interior Girder
        'Truck type 3
        TruckType3(100 * dblDFsi * (1 + dblIFGB), "Girder", "Interior", "Shear") 'Shear force
        TruckType3(100 * dblDFmi * (1 + dblIFGB), "Girder", "Interior", "Moment") 'Moment
        'Truck type 3_2
        TruckType3_2(100 * dblDFsi * (1 + dblIFGB), "Girder", "Interior", "Shear") 'Shear force
        TruckType3_2(100 * dblDFmi * (1 + dblIFGB), "Girder", "Interior", "Moment") 'Moment
        'Truck type 3_3
        TruckType3_3(100 * dblDFsi * (1 + dblIFGB), "Girder", "Interior", "Shear") 'Shear force
        TruckType3_3(100 * dblDFmi * (1 + dblIFGB), "Girder", "Interior", "Moment") 'Moment
        'Truck type legal lane loading
        TruckTypeLL(100 * dblDFsi * (1 + dblIFGB), "Girder", "Interior", "Shear") 'Shear force
        TruckTypeLL(100 * dblDFmi * (1 + dblIFGB), "Girder", "Interior", "Moment") 'Moment

        'Live load analysis - Exterior Girder
        'Truck type 3
        TruckType3(100 * dblDFse * (1 + dblIFGB), "Girder", "Exterior", "Shear") 'Shear force
        TruckType3(100 * dblDFme * (1 + dblIFGB), "Girder", "Exterior", "Moment") 'Moment
        'Truck type 3_2
        TruckType3_2(100 * dblDFse * (1 + dblIFGB), "Girder", "Exterior", "Shear") 'Shear force
        TruckType3_2(100 * dblDFme * (1 + dblIFGB), "Girder", "Exterior", "Moment") 'Moment
        'Truck type 3_3
        TruckType3_3(100 * dblDFse * (1 + dblIFGB), "Girder", "Exterior", "Shear") 'Shear force
        TruckType3_3(100 * dblDFme * (1 + dblIFGB), "Girder", "Exterior", "Moment") 'Moment
        'Truck type legal lane loading
        TruckTypeLL(100 * dblDFse * (1 + dblIFGB), "Girder", "Exterior", "Shear") 'Shear force
```

```
        TruckTypeLL(100 * dblDFme * (1 + dblIFGB), "Girder", "Exterior", "Moment") 'Moment
        'Maximum shear and moment for each load case
        'Interior Girder
        dblMax_V3iGB = Maximum(dblV3iGB) : dblMax_M3iGB = Maximum(dblM3iGB)
        dblMax_V3_2iGB = Maximum(dblV3_2iGB) : dblMax_M3_2iGB = Maximum(dblM3_2iGB)
        dblMax_V3_3iGB = Maximum(dblV3_3iGB) : dblMax_M3_3iGB = Maximum(dblM3_3iGB)
        dblMax_VLLiGB = Maximum(dblVLLiGB) : dblMax_MLLiGB = Maximum(dblMLLiGB)
        'Exterior Girder
        dblMax_V3eGB = Maximum(dblV3eGB) : dblMax_M3eGB = Maximum(dblM3eGB)
        dblMax_V3_2eGB = Maximum(dblV3_2eGB) : dblMax_M3_2eGB = Maximum(dblM3_2eGB)
        dblMax_V3_3eGB = Maximum(dblV3_3eGB) : dblMax_M3_3eGB = Maximum(dblM3_3eGB)
        dblMax_VLLeGB = Maximum(dblVLLeGB) : dblMax_MLLeGB = Maximum(dblMLLeGB)
        'Strength calculation
        'Shear strength calculation
        ShearStrengthGB("Interior") : ShearStrengthGB("Exterior")
        'Moment strength calculation
        MomentStrengthGB("Interior") : MomentStrengthGB("Exterior")
        'Rating Factor Calculation
        'Middle strip
        'Shear RF
        dblRFV3iGB = (dblVriGB - dblVdiGB(0) - dblVddiGB(0)) / dblMax_V3iGB
        dblRFV3_2iGB = (dblVriGB - dblVdiGB(0) - dblVddiGB(0)) / dblMax_V3_2iGB
        dblRFV3_3iGB = (dblVriGB - dblVdiGB(0) - dblVddiGB(0)) / dblMax_V3_3iGB
        dblRFVLLiGB = (dblVriGB - dblVdiGB(0) - dblVddiGB(0)) / dblMax_VLLiGB
        'Moment RF
        dblRFM3iGB = (dblMriGB - dblMdiGB(10) - dblMddiGB(10)) / dblMax_M3iGB
        dblRFM3_2iGB = (dblMriGB - dblMdiGB(10) - dblMddiGB(10)) / dblMax_M3_2iGB
        dblRFM3_3iGB = (dblMriGB - dblMdiGB(10) - dblMddiGB(10)) / dblMax_M3_3iGB
        dblRFMLLiGB = (dblMriGB - dblMdiGB(10) - dblMddiGB(10)) / dblMax_MLLiGB
        'Edge strip
        'Shear RF
        dblRFV3eGB = (dblVreGB - dblVdeGB(0) - dblVddeGB(0)) / dblMax_V3eGB
        dblRFV3_2eGB = (dblVreGB - dblVdeGB(0) - dblVddeGB(0)) / dblMax_V3_2eGB
        dblRFV3_3eGB = (dblVreGB - dblVdeGB(0) - dblVddeGB(0)) / dblMax_V3_3eGB
        dblRFVLLeGB = (dblVreGB - dblVdeGB(0) - dblVddeGB(0)) / dblMax_VLLeGB
        'Moment RF
        dblRFM3eGB = (dblMreGB - dblMdeGB(10) - dblMddeGB(10)) / dblMax_M3eGB
        dblRFM3_2eGB = (dblMreGB - dblMdeGB(10) - dblMddeGB(10)) / dblMax_M3_2eGB
        dblRFM3_3eGB = (dblMreGB - dblMdeGB(10) - dblMddeGB(10)) / dblMax_M3_3eGB
        dblRFMLLeGB = (dblMreGB - dblMdeGB(10) - dblMddeGB(10)) / dblMax_MLLeGB

        dblRFGB = Math.Min(Math.Min(Math.Min(Math.Min(dblRFV3iGB, dblRFV3eGB),
Math.Min(dblRFV3_2iGB, dblRFV3_2eGB)), Math.Min(Math.Min(dblRFV3_3iGB, dblRFV3_3eGB), _
Math.Min(dblRFVLLiGB, dblRFVLLeGB))), Math.Min(Math.Min(Math.Min(dblRFM3iGB, dblRFM3eGB),
Math.Min(dblRFM3_2iGB, dblRFM3_2eGB)), Math.Min(Math.Min(dblRFM3_3iGB, dblRFM3_3eGB), _
Math.Min(dblRFMLLiGB, dblRFMLLeGB))))

        DisplayResultsGB()
    End Sub
    Private Sub msRunGB_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
msRunGB.Click
        RunGB()
    End Sub

    Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AboutToolStripMenuItem.Click
        AboutBox1.ShowDialog()
    End Sub

    Private Sub NewToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NewToolStripMenuItem.Click
        Dim objNew As New Bridge_Strength_Evaluation.MainForm
        objNew.Show()
    End Sub
End Class
```

Declaration

I, the undersigned, declare that the thesis is my original work, has not been presented for a degree in any other university and that all sources of materials used for the thesis have been dully acknowledged.

Name: Yonnas Bayou

Signature: _______________

Place: Faculty of Technology
Addis Ababa University

Date of Submission: __________________