

**Assignment Cover Sheet**

|  |  |  |
| --- | --- | --- |
| **Qualification** | | **Module Number and Title** |
| Higher Diploma in Computing and Software  Engineering | | CSE 4002  Fundamentals in Programming |
| **Student Name & No.** | | **Assessor** |
|  | | Vindya Karunarathne |
| **Hand over date** | | **Submission Date** |
| Xx.xx.xxxx | | xx.xx.xxxx |
| **Assessment type** | **Duration/Length of Assessment Type** | **Weighting of Assessment** |
| **Coursework** | Report and Software  Submission  (3000 words) | 100% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Learner declaration** | | | | |
|  | | | | |
| I, …………………………………………. | | <name of the student and registration number> | , | |
|  | certify that the work submitted for this assignment is my own and research sources are fully | | |  |
| acknowledged. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Marks Awarded** |  | | |
| First assessor |  |  | |
| IV marks |  |  | |
| Agreed grade |  |  | |
| Signature of the assessor |  | Date |  |

**FEEDBACK FORM**

**INTERNATIONAL COLLEGE OF BUSINESS & TECHNOLOGY**

**Module :** CSE 4002  **Student :**

**Assessor : Vindya Karunarathne**

**Assignment :** FP

**Strong features of your work:**

**Areas for improvement:**

**Marks Awarded:**

**Course Work and Practical Assessment**

This assignment is worth 100% of the overall assessment for this module.

# Learning outcomes covered

LO1. Explain structured programming concepts

LO2. Design a basic structured computer program

LO3. Developed a modularized computer programme for a prepared design

LO4. Compile software testing and documentation

## Scenario

**BestMeal** is a newly built Meal kit restaurant situated in west London. This restaurant is famous for their consistently excellent recipes with quality ingredients.

Responding to rapid customer growth, management has decided automate the entire business process. Initial step BPA (business process automation) team have analyzed the business process and planned to automate the Meal kit ordering process.

Assume that you are a group member of **BestMeal** business process automation team.

Design and develop the Meal kit-ordering module for **BestMeal** Restaurant automation.

**Main requirements**

* Manage available Meal kits
* View available Meal kits.
* Search specific Meal kit.

**Other requirements are.**

* User login.
* Logout.
* Exit
* Help

**Note that: Refer the following link to get an idea https://www.homechef.com**

Create an error free, effective one with user-friendly interfaces, appropriate messages, outputs and menu driven application using C++ programming language. Students can use appropriate data structures and text files to store information.

**Carefully investigate the given scenario and provide the proposed solution.**

**Attach softcopy of error free program with your documentation.**

**Keep all the backups**

## TASKS Task 01 – Programming methodology (LO1) -Report

Describe the usage of basic programming methodology such as controlling structures (selection, repetition) and modularization.

• Provide appropriate examples. **(10 marks)**

## Task 02 - System Design (LO2) -Report

Identify the requirements of the system to be implemented and **produce detailed requirements specification. Design flow charts** according to the SRS.

• Use appropriate modularization to reduce the complexity of the design*.* **(20 marks)**

## Task 03 – System Implementation (LO3) - Software submission

Implement a functional C++ program to meet the requirements given in the specification, by following the design created above. **(40 marks)**

## Task 04 - System Testing and Documentation (LO4) -Report

Produce proper testing documentation including test plan, test cases and test results. Conduct user acceptance testing and provide feedback with sample questionnaires used.

• Proper **standard documentation** need to be followed throughout the report and language need to be used accordingly. **(30 marks)**

## Assessment Criteria

**Task 01 –** Programming Methodology **(LO1)**

|  |  |  |
| --- | --- | --- |
| **This submission will be assessed as follows**  **Criteria** | **Total marks**  **Allocated** | **Marks obtained by the student for the answer provided** |
| **Out of 10** |
| **Excellent** Understanding   * Outstanding/Excellent answer given with strong discussion providing appropriate examples * Excellent knowledge in basic programming methodology * Evidence of Research and proper use of   referencing | 8-10 |  |
| **Good** Understanding   * Comprehensive answer given with very good understanding providing appropriate examples * Good knowledge in basic programming methodology | 6-7 |  |
| **Satisfactory** Understanding   * A satisfactory answer with a demonstration of basic understanding of the programming methodology * Very basic answer with limited examples | 4-5 |  |
| **Poor** Understanding   * lack of understanding about basic programming concepts * Task is not attempted or attempted with an insufficient, incomplete answer | 0-3 |  |

**Task 02 -** System Design **(LO2)**

|  |  |  |
| --- | --- | --- |
| **This submission will be assessed as follows**  **Criteria** | **Total marks**  **Allocated** | **Marks obtained by the student for the answer provided** |
| **Out of 20** |
| **Excellent** Design   * Excellent SRS given in detail * Highly **detailed** diagram * **Use of modularization** concepts clearly visible * **Excellent use of symbols** * **Clarity and Reduce complexity** of the design * Backed by relevant assumptions | 15-20 |  |
| **Good** Design   * Detail **SRS** including functional and nonfunctional requirements, data and file structure requirements * **Flow charts** following standard notations in flow charting **and pseudo codes** using proper structured English * **Accurate use** of selection repetition structures * **Logical** and continuous flow of instructions along the design | 13-14 |  |
| **Satisfactory** Design   * Basic **SRS** including functional requirements * **Clear identification** and **application** of symbols in flow charts * Average level design diagrams given | 9-12 |  |
| **Poor** Design   * Evidence of lack of understanding systems requirement specification * Poor use of design tools and symbols * Design diagrams with invalid flows, incomplete diagrams with logical errors | 0-8 |  |

**Task 03** – System Implementation (LO3)

|  |  |  |
| --- | --- | --- |
| **This submission will be assessed as follows**  **Criteria** | **Total marks**  **Allocated** | **Marks obtained by the student for the answer provided** |
| **Out of 40** |
| **Excellent** implementation   * Excellent use of **control structures** with improved coding efficiency * Use **file handling** techniques for storage and backup requirements * **Excellent Modularization** with **effective data passing** between developed modules. * Appropriate **arrays, structs(records)** used * Easy navigation between modules, **accuracy, creativity and completeness** of the system | 29-40 |  |
| **Good** implementation   * Use of **Comments** to improve code readability * good use of **control structures** with proper understanding * **Modularize** according to the given design. * Use of input validations, onscreen help options and User friendliness of the system | 25-28 |  |
| **Satisfactory** implementation   * **Operational system** according to the   requirements of the scenario   * Average use of **data types** and **operators** * Average use of **control structures** (selection and repetition) | 17-24 |  |
| **Poor** implementation   * Poor implementation with syntax errors * Lack of knowledge of the language basics used * Cannot fulfill basic system requirements | 0-16 |  |

**Task 04** - System Testing and Documentation (LO4)

|  |  |  |
| --- | --- | --- |
| **This submission will be assessed as follows**  **Criteria** | **Total marks**  **Allocated** | **Marks obtained by the student for the answer provided** |
| **Out of 30** |
| **Excellent** Documentation   * Excellent Test documentation with detail **test plan** and **test cases** * Acceptance test with proper questionnaire samples.Well analyzed user feedback which supports recommendations. * **Testing conclusion** with critical review and future recommendations * Appropriate use of language and Standard report format followed * Proper use of Referencing | 22-30 |  |
| **Good** Documentation   * Detailed **Test Plan** * Appropriate **Test Cases** * Acceptance test with **User feedback** and test **conclusion** * Good documentation | 19-21 |  |
| **Satisfactory** Documentation   * Basic **Test Plan** * Average **Test Cases** * Average documentation | 13-18 |  |
| **Poor** Documentation   * Lack of test plan, poor test cases * No proper evidence of testing * Poor report formatting | 0-12 |  |

|  |  |  |
| --- | --- | --- |
| **Total Marks** | **100** |  |

**Submission Guidelines**

|  |  |  |
| --- | --- | --- |
|  | Submission format | Report |
|  | Paper Size: | A4 |
|  | Words: | 3000 words |
|  | Printing Margins: | LHS; RHS: 1 Inch |
|  | Header and Footer: | 1 Inch |
|  | Basic Font Size: | 12 |
|  | Line Spacing: | 1.5 |
|  | Font Style: | Times New Roman |

 **Referencing should be done strictly using Harvard system**

# Acknowledgement

This report is supported by HND in Computing and Software Engineering. I have been supported by many people to conclude this report effectually. I wish to express my sincere gratitude to miss. Vindya Karunarathne lecturer who helped extensively, offered indispensable assistance, support and guidance. Also I wish to give my thanks to my friends as giving more invaluable assistance. I wish to express my love and gratitude to my beloved family, for their understanding and unbounded love and support, along the duration of my assignment.

Thank you,

Contents

[Learning outcomes covered 3](#_Toc61035186)

[Scenario 3](#_Toc61035187)

[TASKS Task 01 – Programming methodology (LO1) -Report 4](#_Toc61035188)

[Task 02 - System Design (LO2) -Report 4](#_Toc61035189)

[Task 03 – System Implementation (LO3) - Software submission 4](#_Toc61035190)

[Task 04 - System Testing and Documentation (LO4) -Report 4](#_Toc61035191)

[Assessment Criteria 5](#_Toc61035192)

[Acknowledgement 10](#_Toc61035193)

[Task 1 - Programming Methodology (LO1) 11](#_Toc61035194)

[1.1 Introduction to Programming 11](#_Toc61035195)

[1.2 Programming Methodologies 11](#_Toc61035196)

[1.2.1 Structured Programming Methodology 12](#_Toc61035197)

[1.2.2 Object-Oriented Programming Methodology 12](#_Toc61035198)

[1.3 Control Structures 12](#_Toc61035199)

[1.3.1 Selection Structures 12](#_Toc61035200)

[1.3.2 Repetition Structures 19](#_Toc61035201)

[1.4 Modularization (Modular Programming) 23](#_Toc61035202)

[Functions in Programming 23](#_Toc61035203)

[Task 2 – System Design (LO2) 26](#_Toc61035204)

[2.1 SRS for Chrystal Clinic Billing System 26](#_Toc61035205)

[2.1.1 Introduction of Chrystal Skin Clinic 26](#_Toc61035206)

[2.1.2 Purpose 26](#_Toc61035207)

[2.1.3 Scope of System 26](#_Toc61035208)

[2.1.4 Requirements 26](#_Toc61035209)

[2.2 Flowcharts 28](#_Toc61035210)

[Task 3 – System Design (LO2) 32](#_Toc61035211)

[Task 4 – Software Testing and Documentation 42](#_Toc61035212)

[4.1 Introduction to Software Testing 42](#_Toc61035213)

[4.2 Software Testing Methodologies 42](#_Toc61035214)

[4.2.1 Functional Testing 42](#_Toc61035215)

[4.2.2 Non-Functional Testing 43](#_Toc61035216)

[4.2.3 Black Box Testing 44](#_Toc61035217)

[4.2.4 White Box Testing 44](#_Toc61035218)

[Test Plan 45](#_Toc61035219)

[References 46](#_Toc61035220)

[References 46](#_Toc61035221)

[Figure 1 15](#_Toc61035222)

[Figure 2 15](#_Toc61035223)

[Figure 3 16](#_Toc61035224)

[Figure 4 16](#_Toc61035225)

[Figure 5 17](#_Toc61035226)

[Figure 6 17](#_Toc61035227)

[Figure 7 18](#_Toc61035228)

[Figure 8 18](#_Toc61035229)

[Figure 9 18](#_Toc61035230)

[Figure 10 19](#_Toc61035231)

[Figure 11 19](#_Toc61035232)

[Figure 12 20](#_Toc61035233)

[Figure 13 20](#_Toc61035234)

[Figure 14 21](#_Toc61035235)

[Figure 15 21](#_Toc61035236)

[Figure 16 22](#_Toc61035237)

[Figure 17 23](#_Toc61035238)

[Figure 18 24](#_Toc61035239)

[Figure 19 24](#_Toc61035240)

[Figure 20 25](#_Toc61035241)

[Figure 21 26](file:///C:\Users\User\Documents\fcp-assignment93%20thilina.docx#_Toc61035242)

[Figure 22 26](#_Toc61035243)

[Figure 23 27](#_Toc61035244)

[Figure 24 27](#_Toc61035245)

[Figure 25coding 1 34](#_Toc61035246)

[Figure 26coding 2 35](#_Toc61035247)

[Figure 27coding 3 36](#_Toc61035248)

[Figure 28coding 4 37](#_Toc61035249)

[Figure 29coding 4 38](#_Toc61035250)

[Figure 30coding 5 39](#_Toc61035251)

[Figure 31coding 6 41](#_Toc61035252)

[Figure 32 46](#_Toc61035253)

# Task 1 - Programming Methodology (LO1)

## Introduction to Programming

As the meaning of programming can be considered as giving set of instructions, commands and syntax in order to execute a specific task. There are many programming languages which contain instructions as a medium to communicate with computers. First programming language was “Machine Level” which used binary or equivalent notations for real numeric values. After that “Assembly language” was used in 1950’s. Correspondingly “High Level Languages”, “System Programming Languages”, “Object Oriented Programming” and “Scripting” was created.

(Anon., 2011)

## Programming Methodologies

Programs are developed for specific tasks. But when it comes to solving problems by the programs it will be complicated. So programming methodology is used to analyze complicated problems, planning for software development and controlling the development process.

It can be divided as follow.

(Anon., n.d.)

### 1.2.1 Structured Programming Methodology

Structured programming method is a software development method which used modularization and structured design. This the method which reduces the errors and interruptions by breaking down large programs or scripts into smaller modules known as functions.

(Anon., 2011)

### 1.2.2 Object-Oriented Programming Methodology

Object-Oriented programming methodology is both data and functions are combined into single unit that operate on that data. This unit is known as an object. The functions of object are known as member functions; it is the only way to access object’s data. If the programmer wants to access the function inside the object, he/she should call the member function in the object.

(Lafore, n.d.)

## Control Structures

Control structure is a statement through which is used to control the operations of the program what kind of function it will perform, when it terminates or continue under a particular circumstance. Control structures mainly can be divided into two parts as discussed in below.

(Anon., 2000)

### 1.3.1 Selection Structures

There are some statements in selection structures.

* If Statement
* If-Else Statement
* Nested If-Else Statement
* Ternary/ Conditional Operator
* Switch-Case Statement

#### 1.3.1.1 If Statement

If statement is the basic statement of the all control structures. If the condition is true, the statement will be executed. If the condition is false, the statement will be terminated or skipped.

(Anon., n.d.)

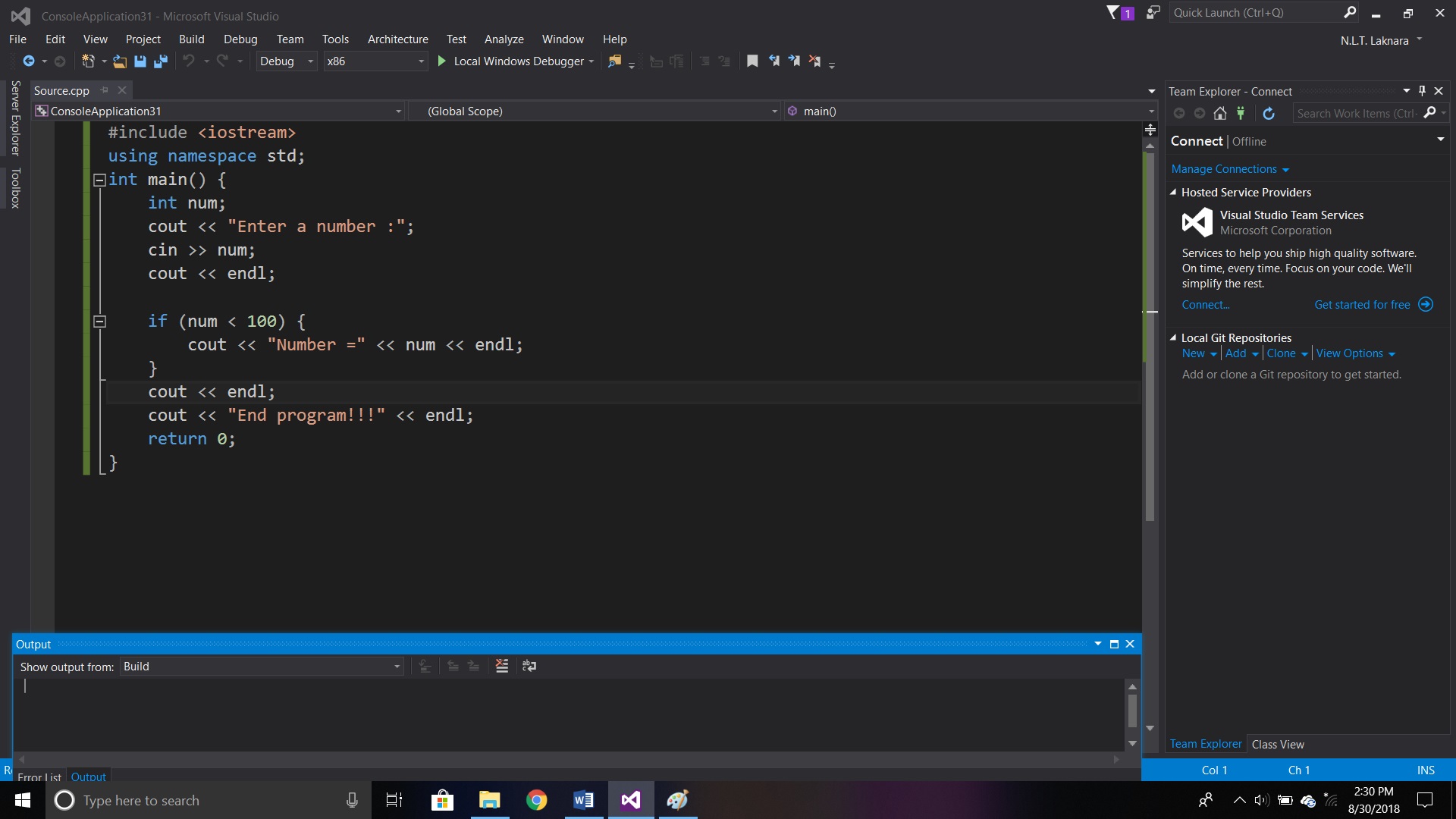


Figure 1

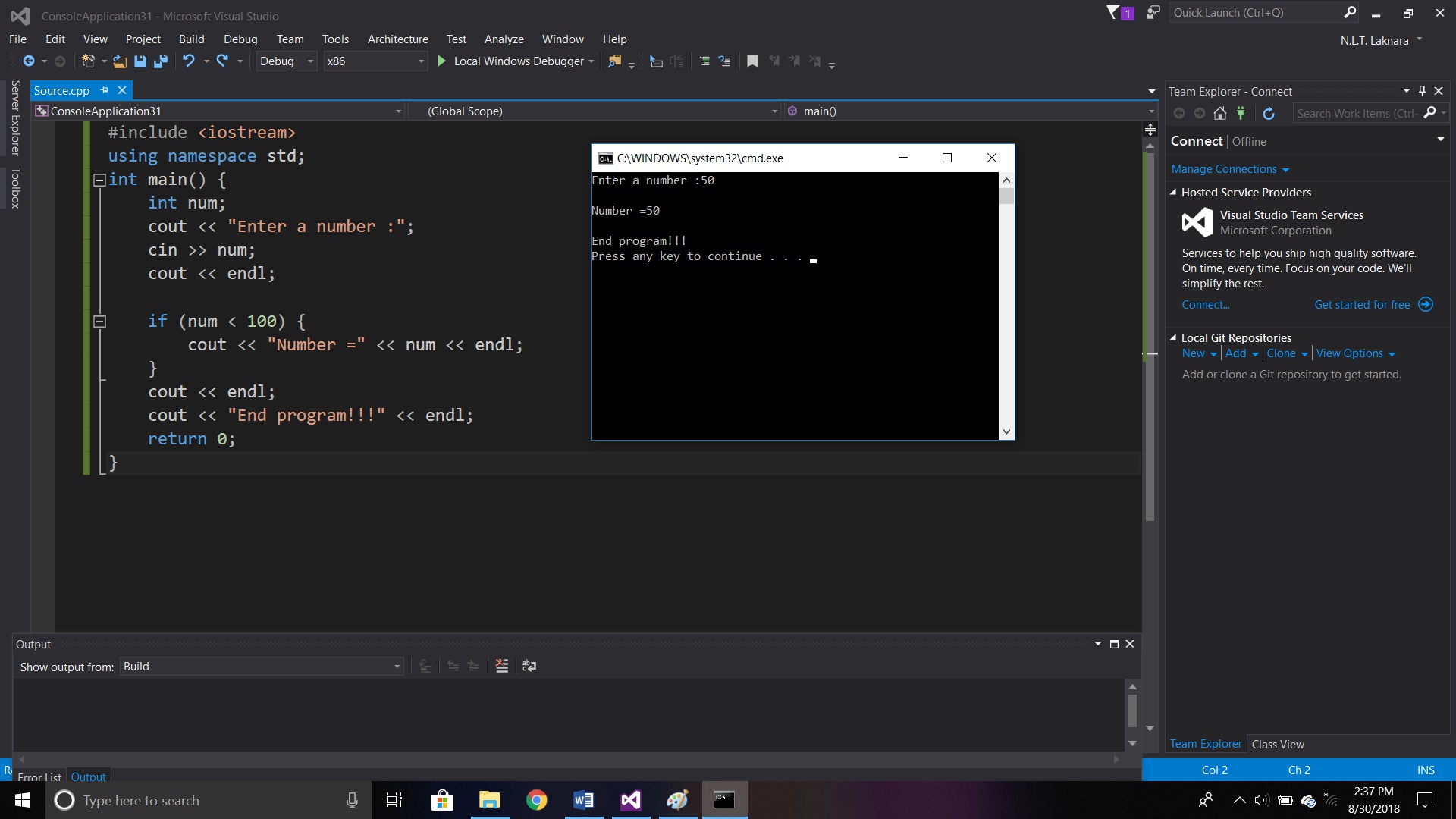


Figure 2

#### 1.3.1.2 If-Else Statement

If-Else statement has another optional statement besides If statement. Inside codes of ‘Else’ statement will be executed if the condition is false. Then codes of the ‘If’ statement will be skipped. If the condition is true, codes of the If statement will be executed and codes of Else statement will be skipped.

(Anon., 2019)

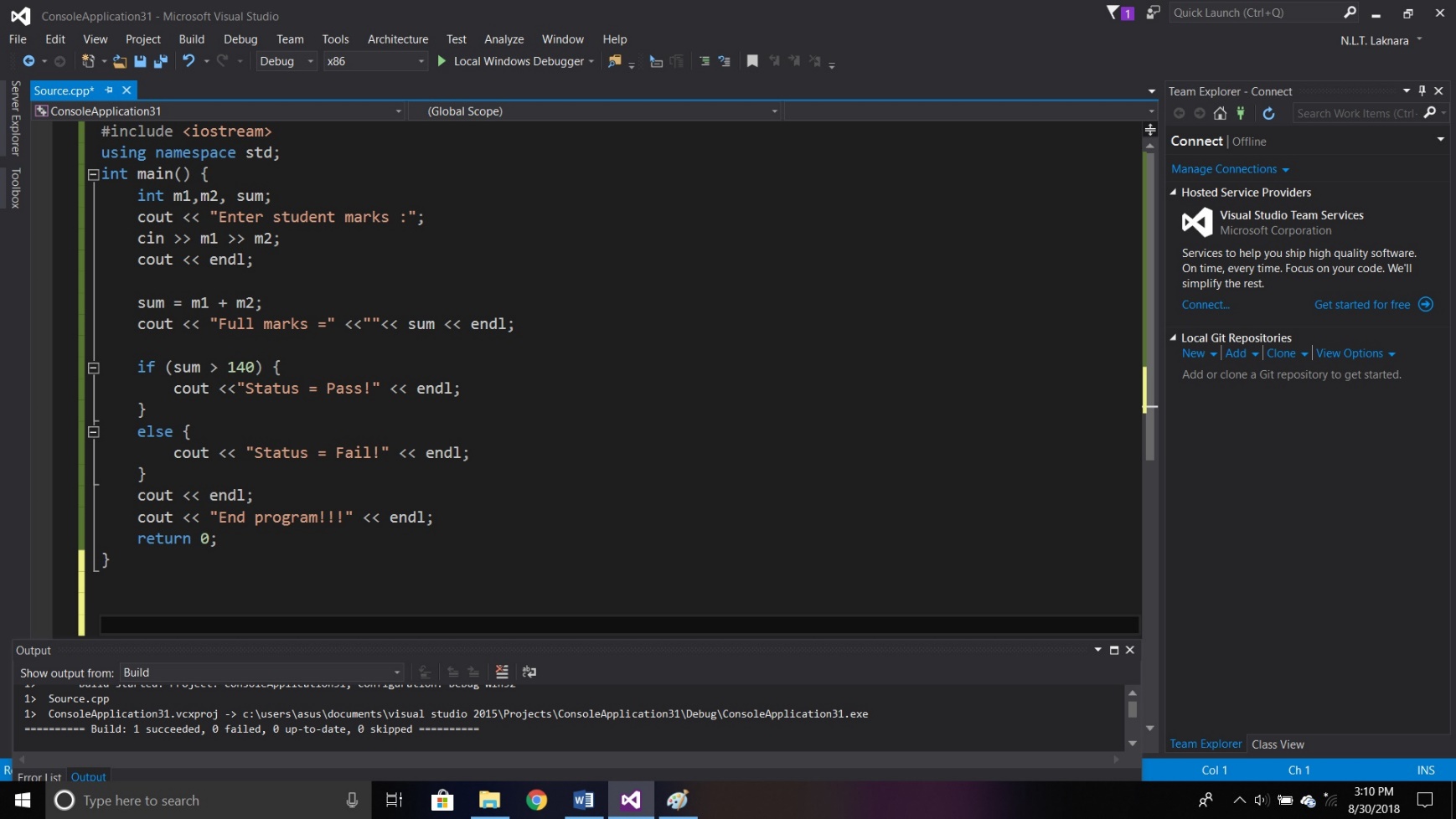
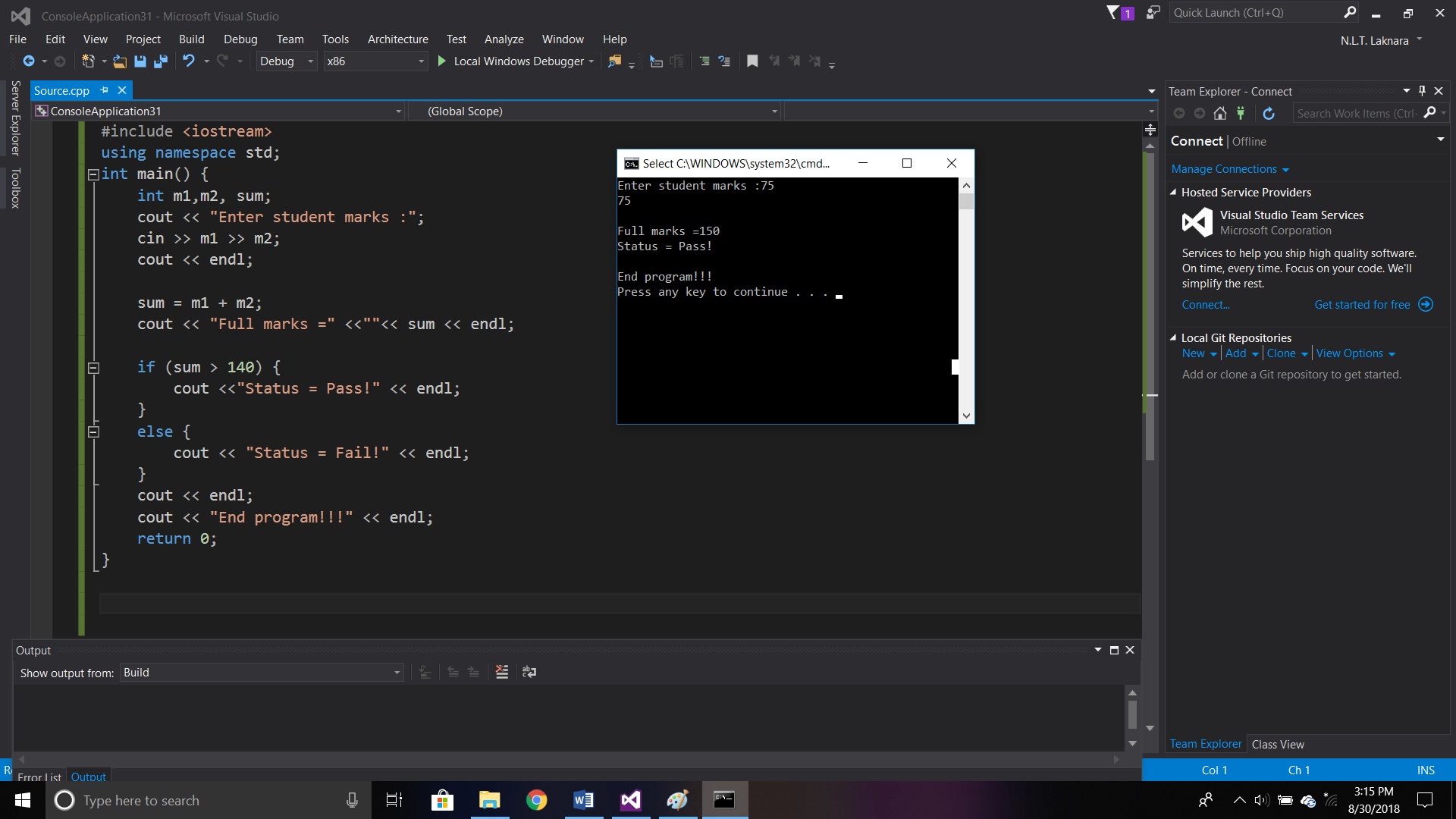


Figure 3

****

* **True statement**

Figure 4

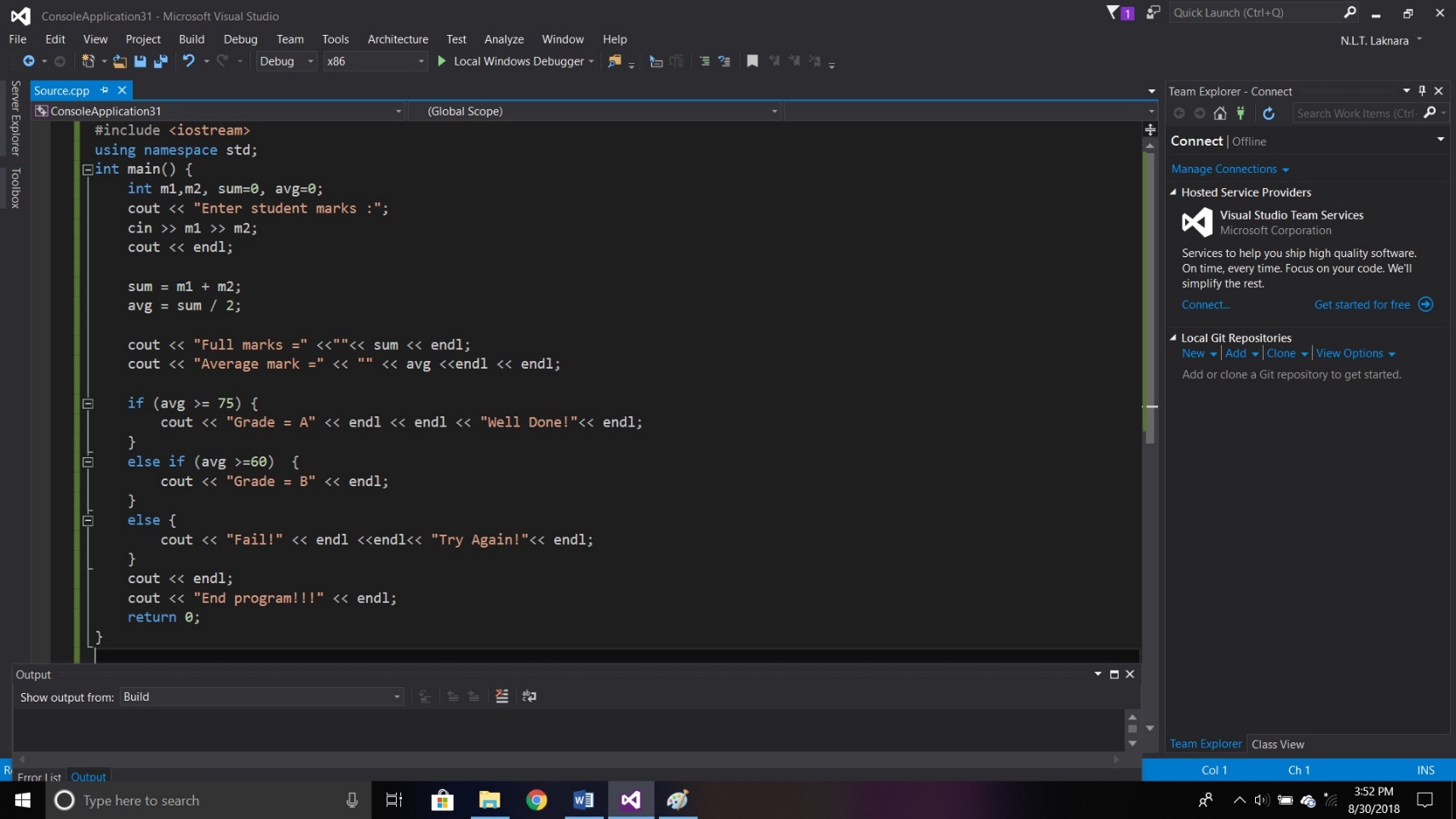


* **False statement**

Figure 5

#### 1.3.1.3 Nested Else-If Statement

If-Else statement is depending on whether the expression is true or false. Perhaps, the choice has to be decide from more than two possibilities.

This statement allows to confirm multiple expressions and execute different codes from more than two conditions.

(Anon., n.d.)

Figure 6

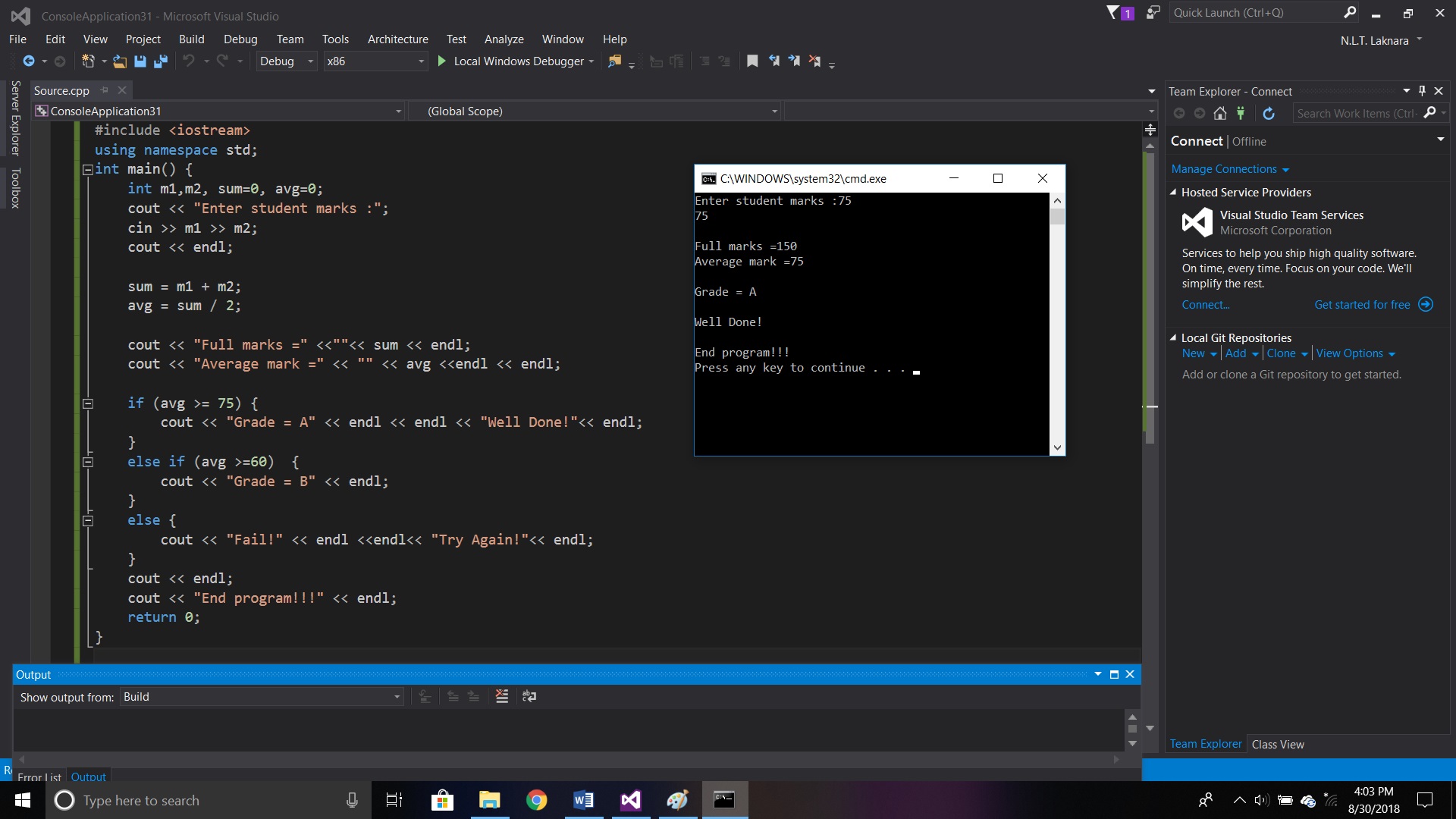
****

Figure 7

****

Figure 8

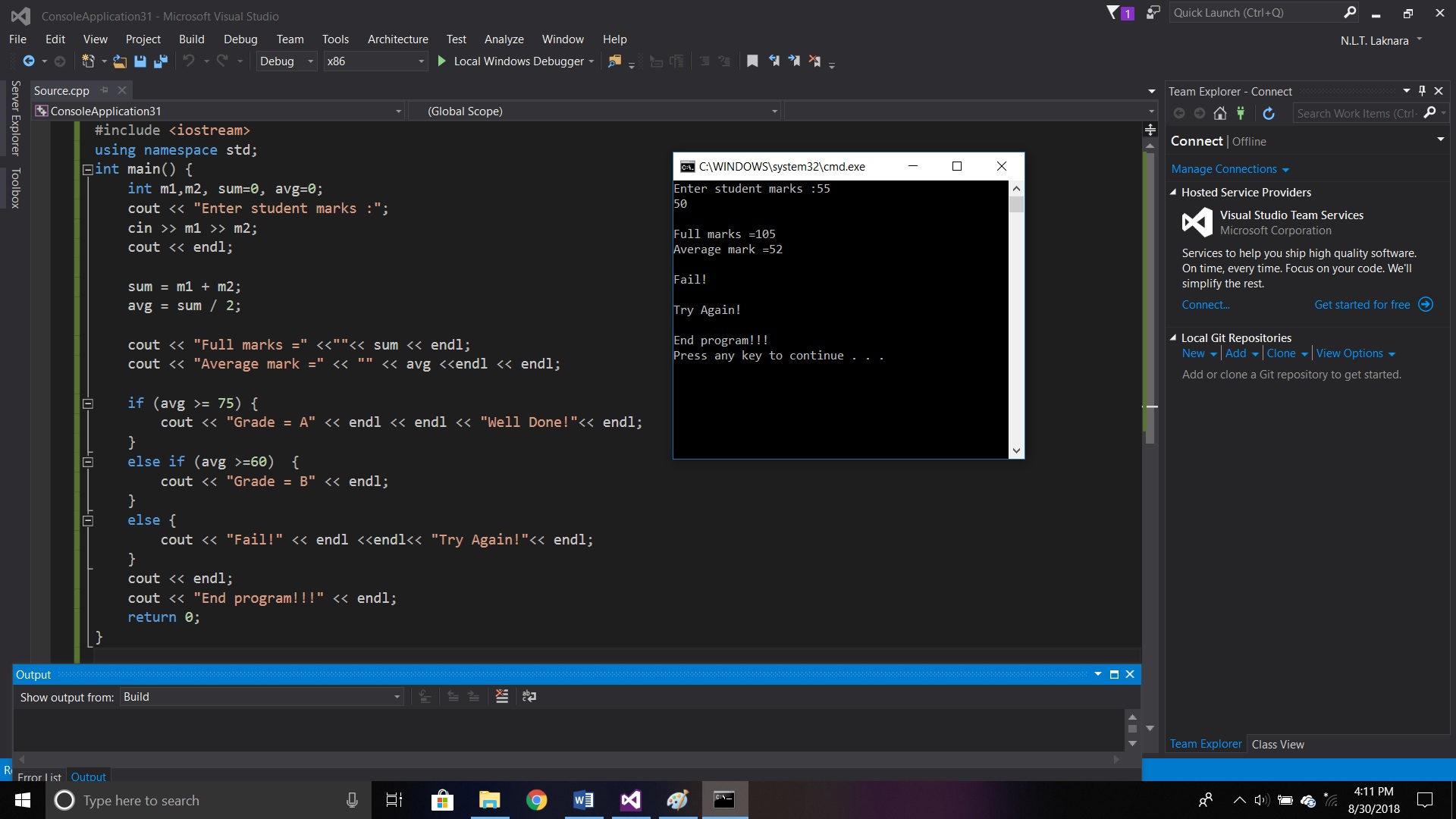


Figure 9

#### 1.3.1.4 Conditional / Ternary Operator

Conditional / Ternary operator is the alternative for if-else statement. A teranry operator has 3 operands.



**Expression 1** is the statement related to true. If the test condition is true, then the result of the **Expression 1** will be returned by the ternary operator. Unless the condition is true, then the **Expression 2** will be returned.

(Anon., n.d.)

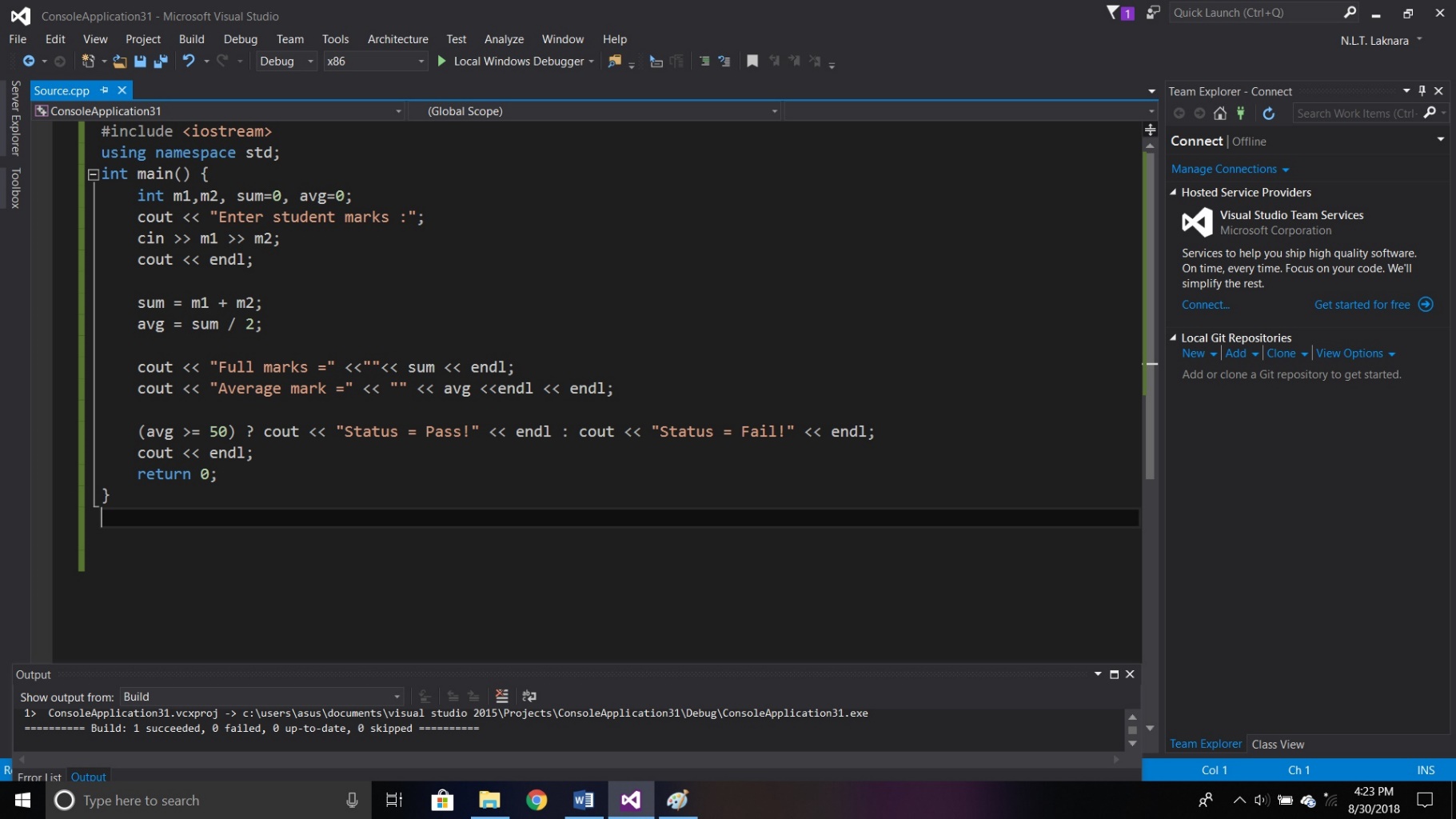


Figure 10

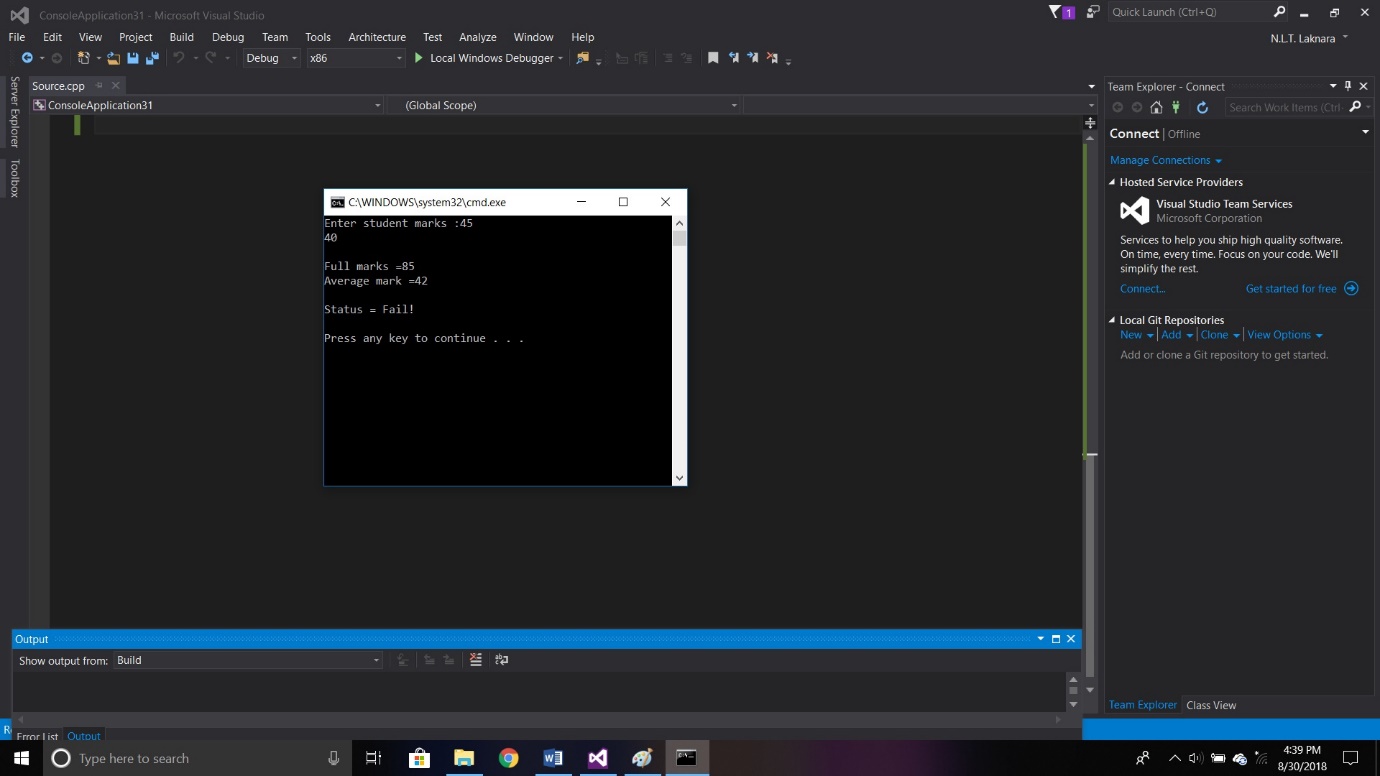


Figure 11

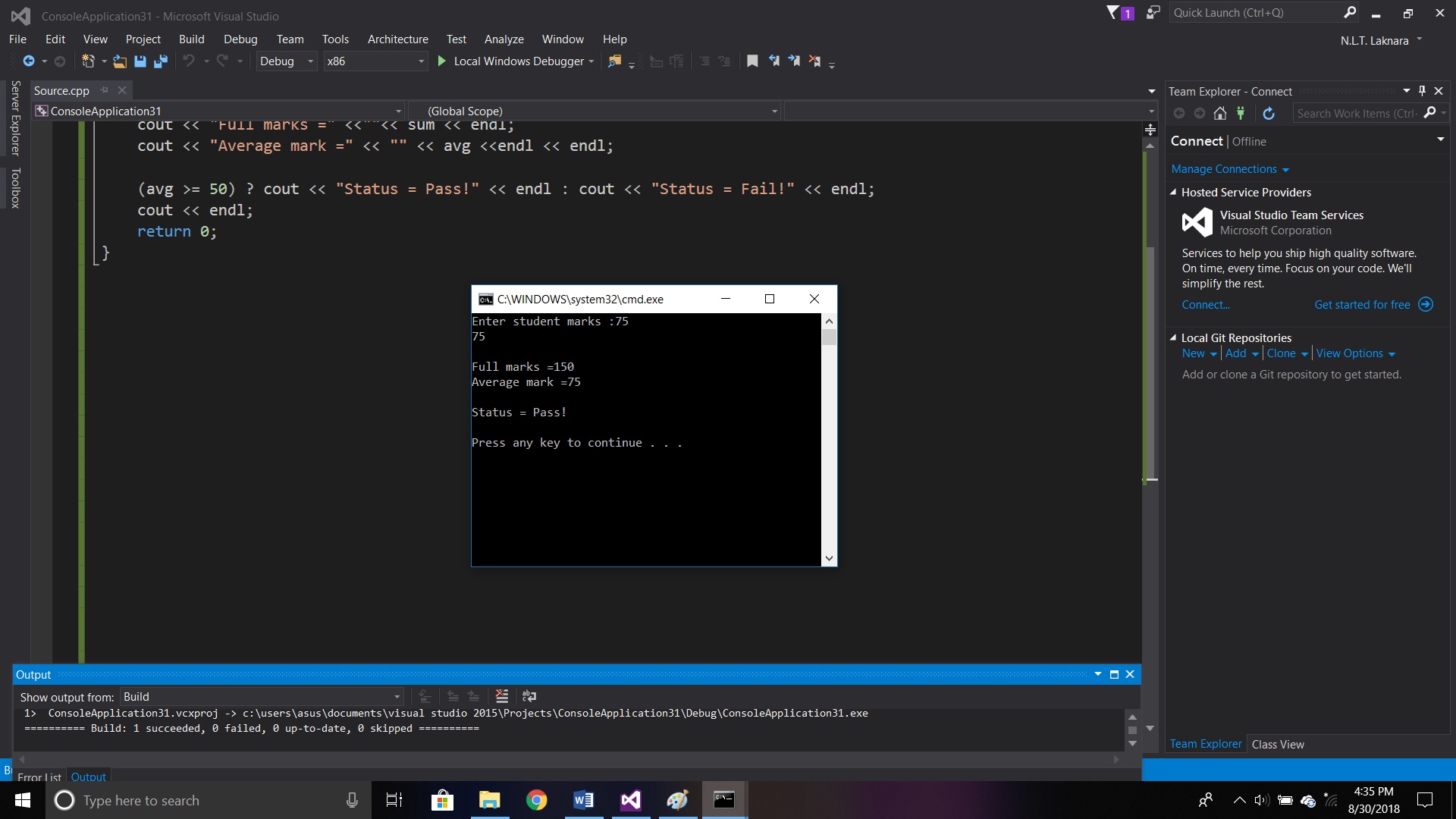


Figure 12

#### 1.3.1.5 Switch Case Statement

In switch case statement allows to check on the value of a single variable in a sequence.

(Anon., n.d.)

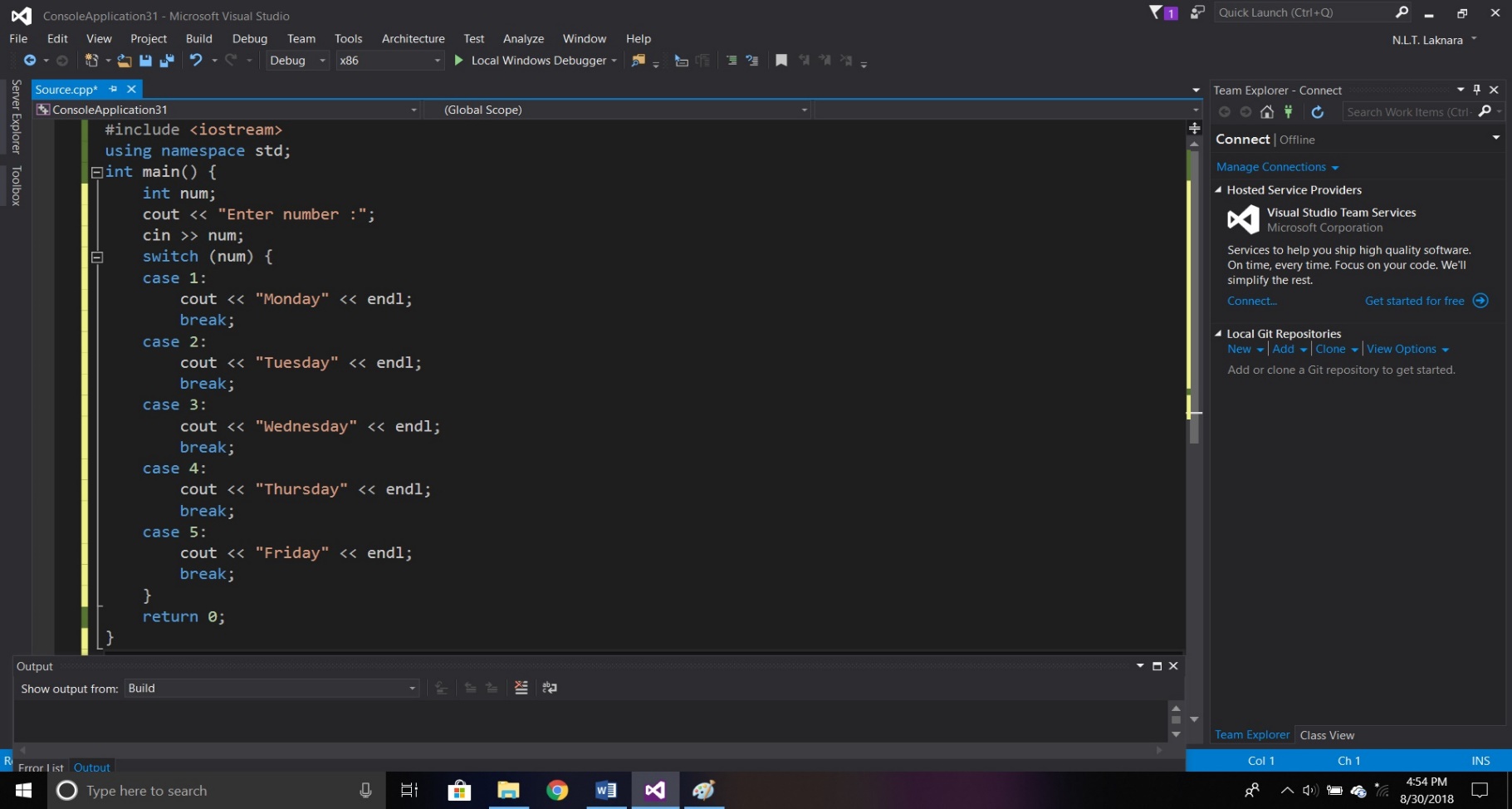


Figure 13

### 1.3.2 Repetition Structures

In programming, repetition structures are used to execute the same action in multiple times. This structures enable through loops.

Three types of loops are available in C++,

* For Loop
* While Loop
* Do-While Loop

(Anon., n.d.)

#### 1.3.2.1 For Loop

For loop is executing repeatedly until the result of the test expression is false.

**Syntax**

****

In this initialization statement will be executed only once at the beginning and then the text expression will be evaluated. If the text expression is false, for loop will terminate. If it is true, until the text expression is false for loop will be repeated.

(Anon., 2012)

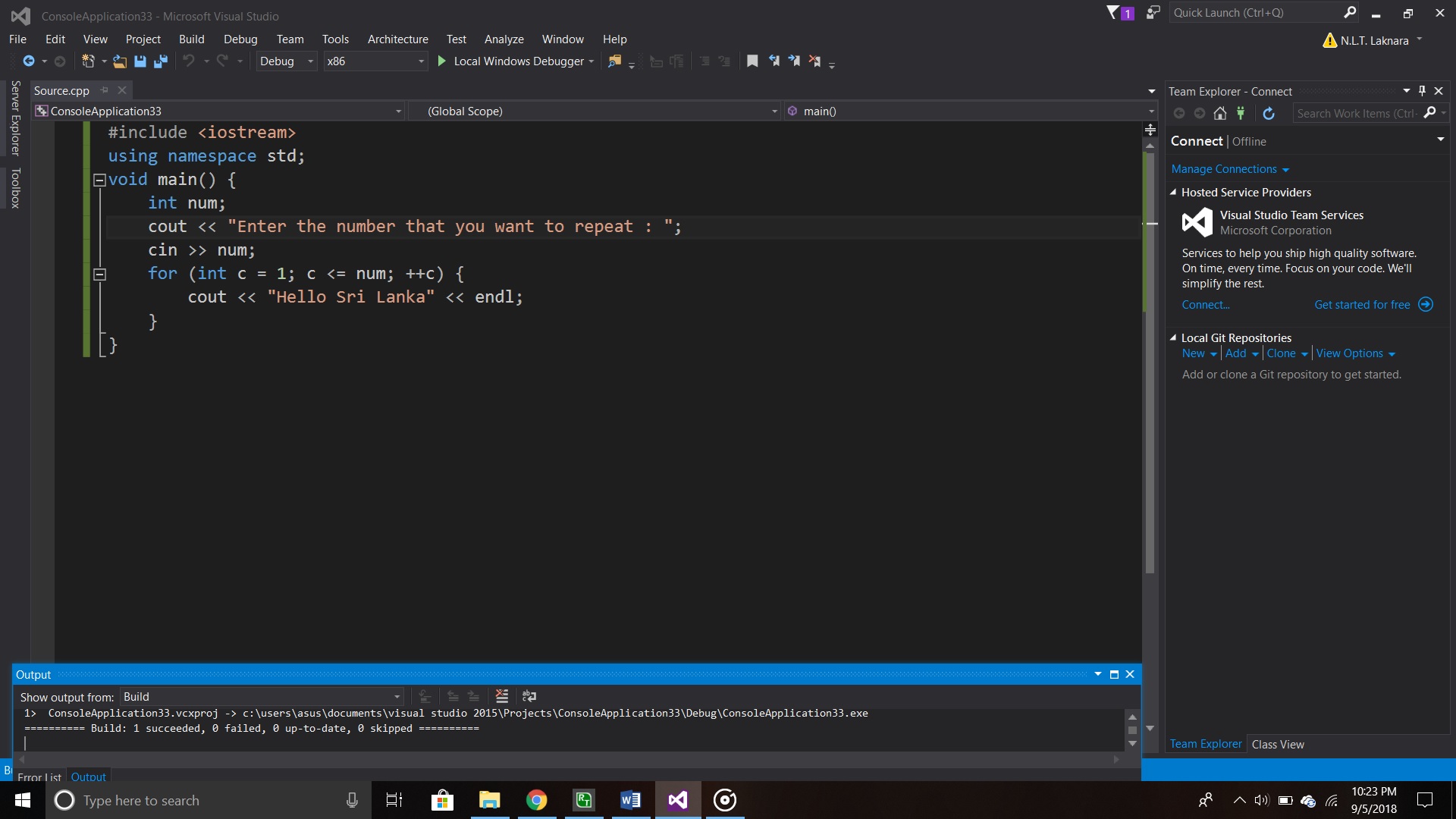
****

Figure 14

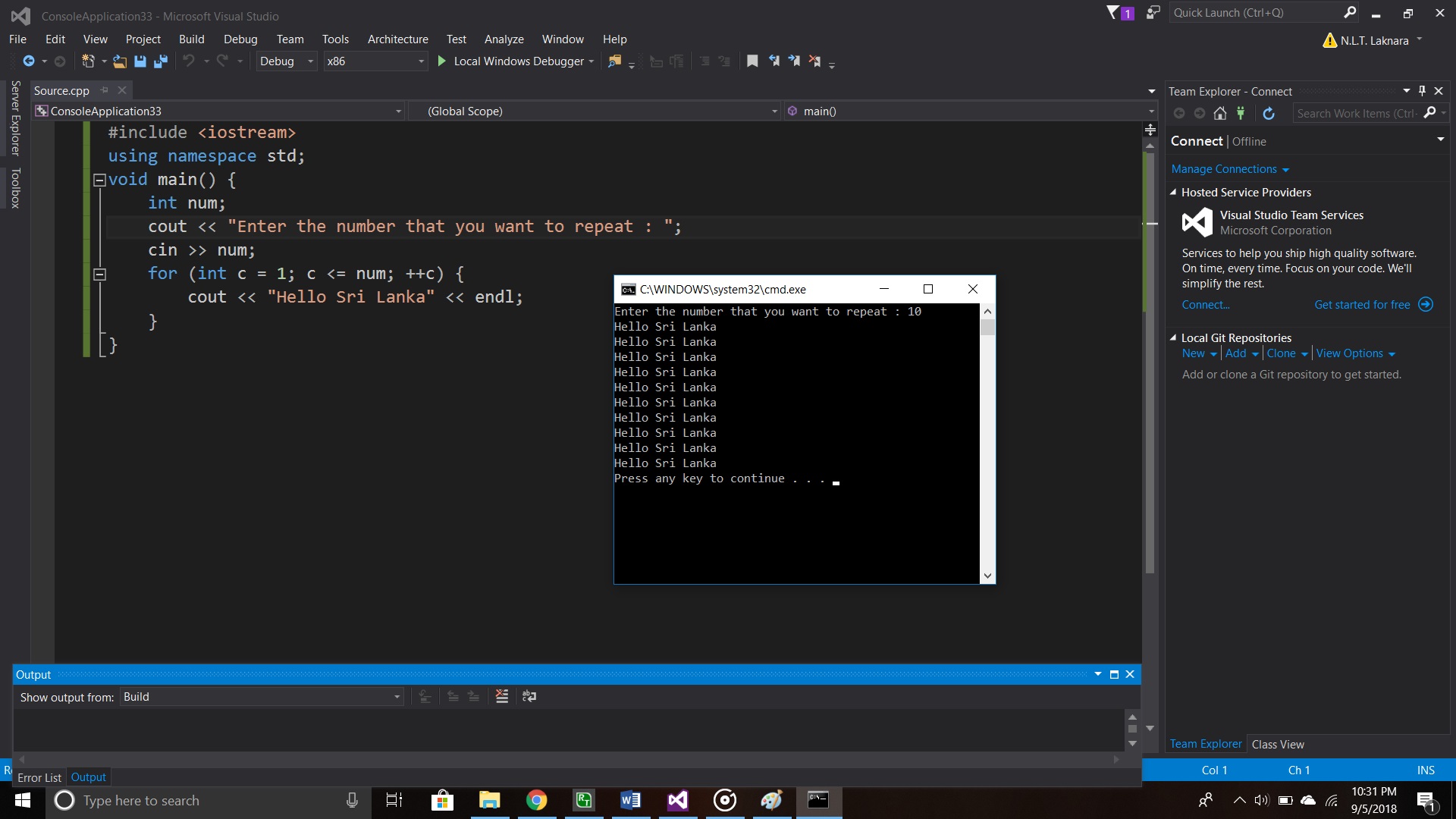
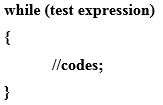


Figure 15

#### 1.3.2.2 While Loop

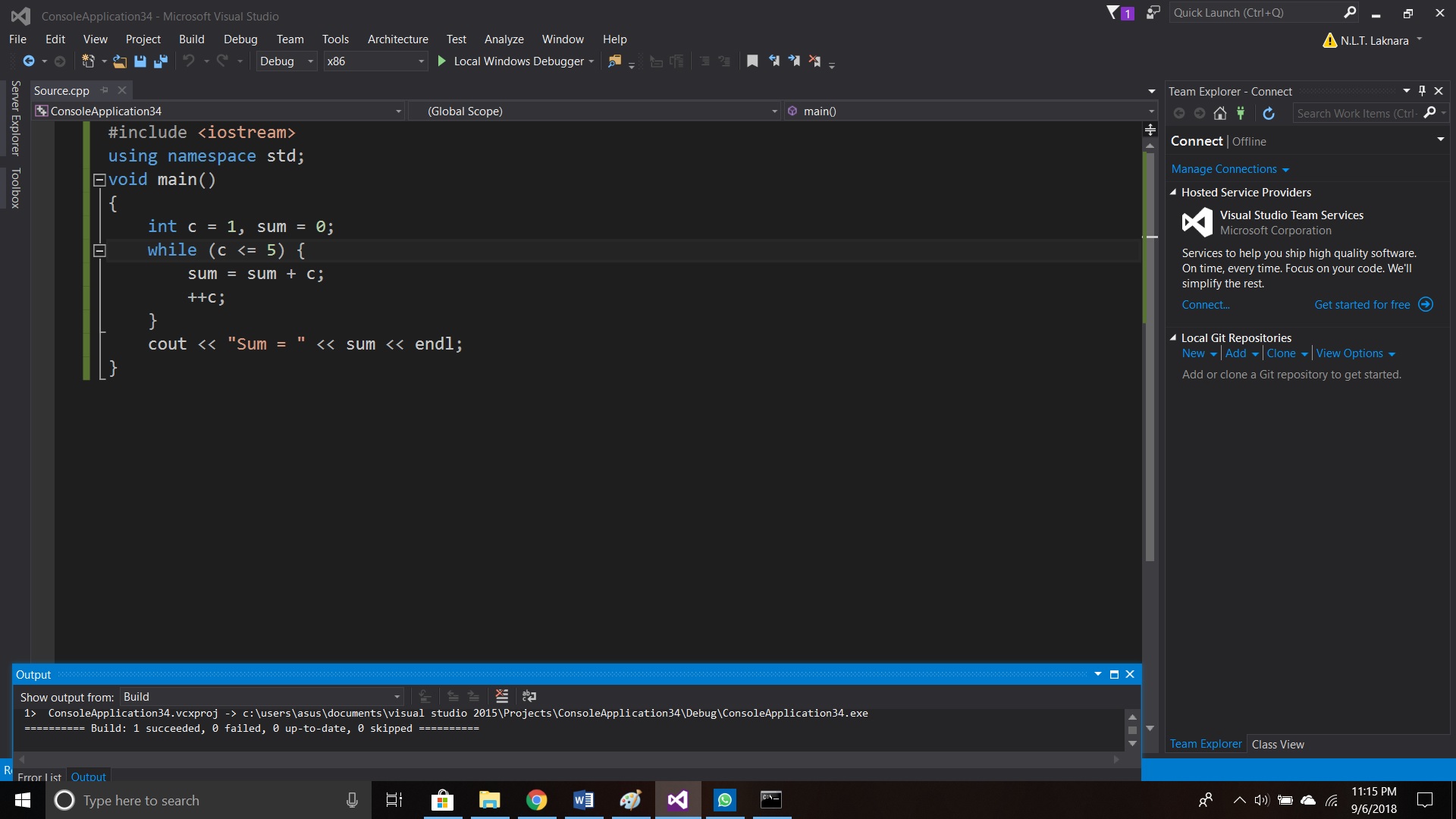
In programming the while loop is used to execute repeatedly until the test expression is false.

**Syntax**



In this loop the test expression will be evaluated first. If it is true, the codes inside the loop will be executed and again test expression will be evaluated repeatedly till the test expression is false. When the test expression is false while loop will be terminated.

(Anon., n.d.)

****Figure 16

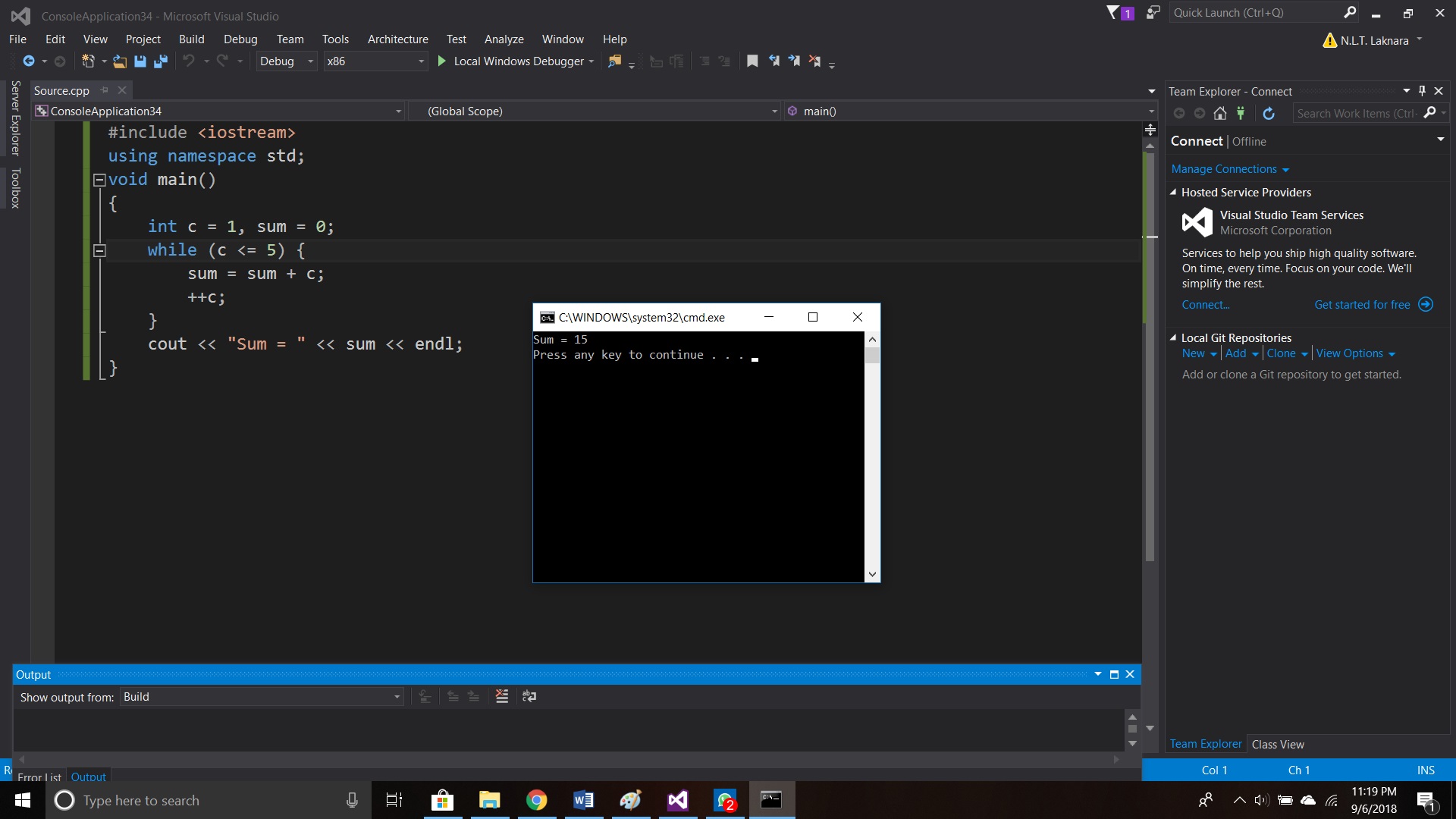
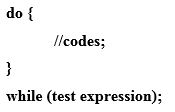
****

Figure 17

#### 1.3.2.3 Do-While Loop

Do while loop is different from while loop within an important difference. The codes of the do while loop is executed before the test expression is evaluated.

**Syntax**

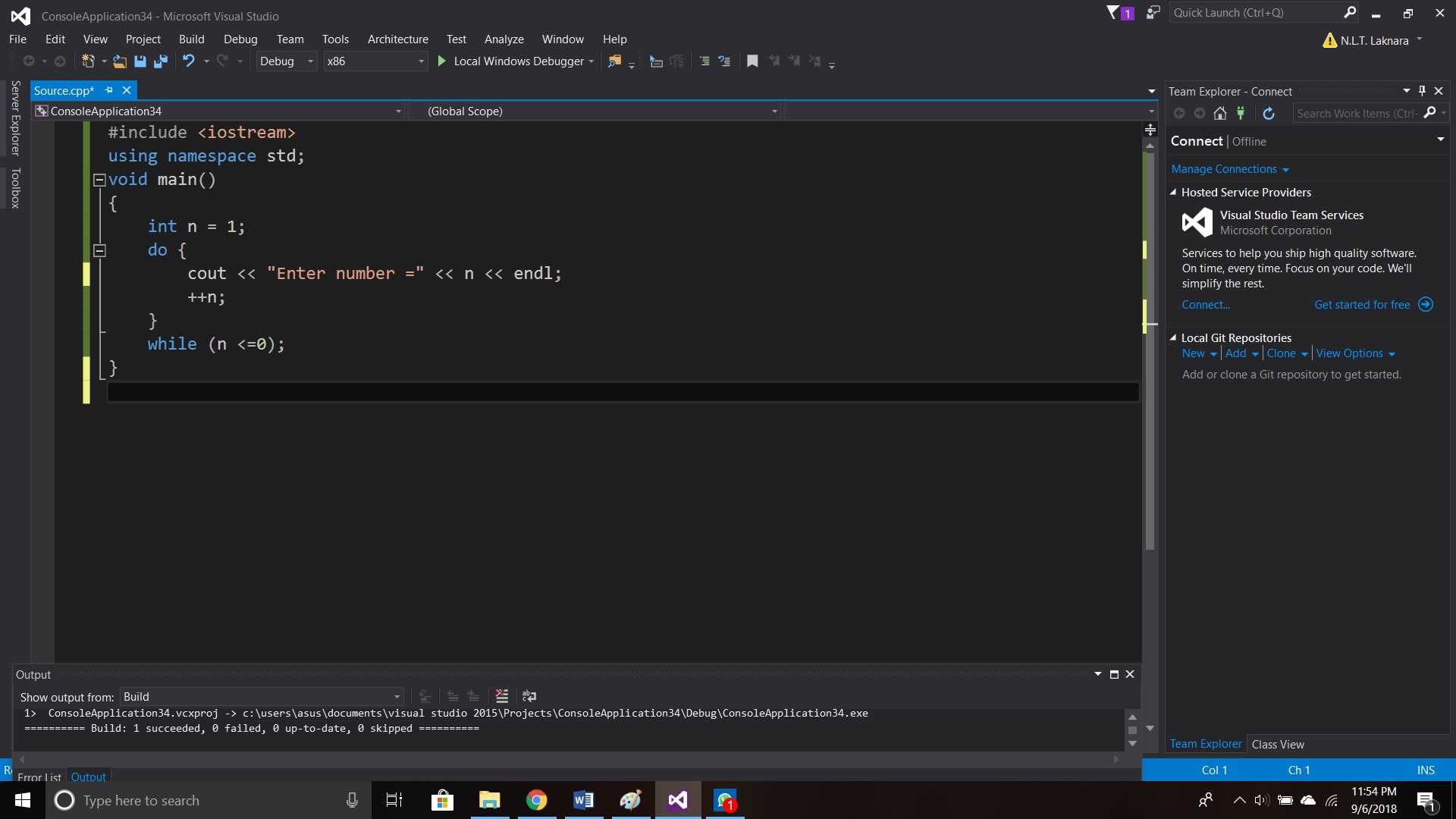
In this loop the codes will be executed at least once whether the test expression is true or false.

Figure 18

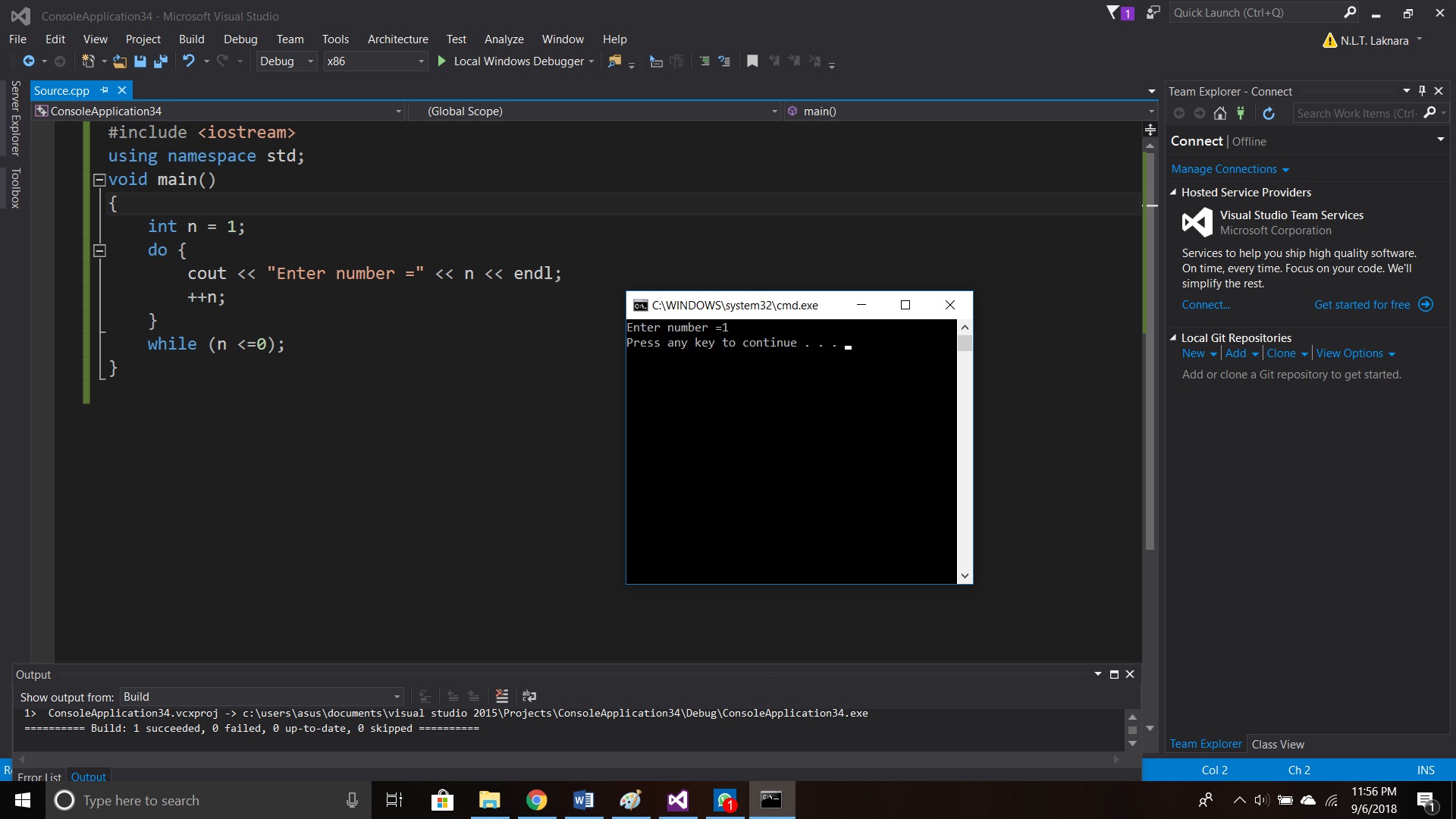


Figure 19

## Modularization (Modular Programming)

Modular programming is used to simplify the problem; the main entire program is subdividing into sub programs. In this modular programming errors can be identified easily and easy to understand.

(Anon., n.d.)

### Functions in Programming

Function is a technique which enables modularization in programming. There are two functions in modularization in programming.

* Library Functions (Pre-defined Functions)
* User-defined Functions

#### Library Functions

These functions are already provided with the programming itself. Library functions can be easily accessed by importing the header file which includes in the library function. There are plenty library functions which is provided by the compiler.

(Anon., 2017)

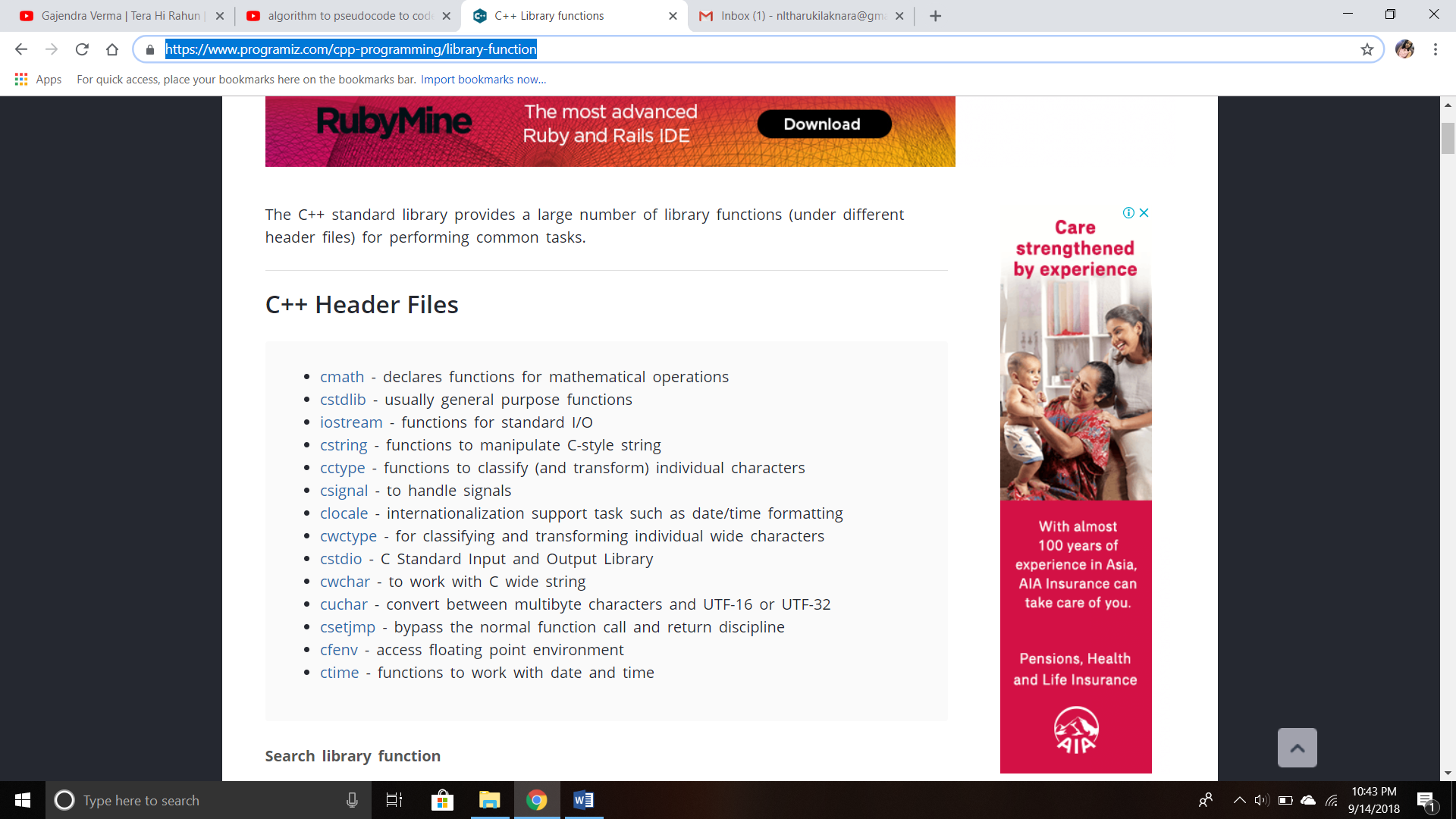
Header Files:

Figure 20

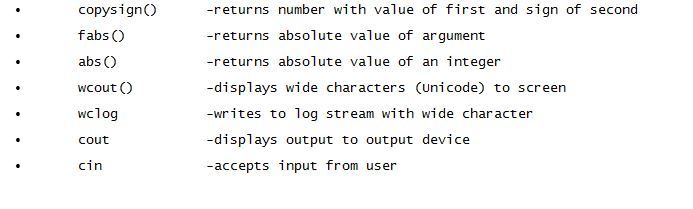
**Library functions:**

Figure 21

(Anon., n.d.)

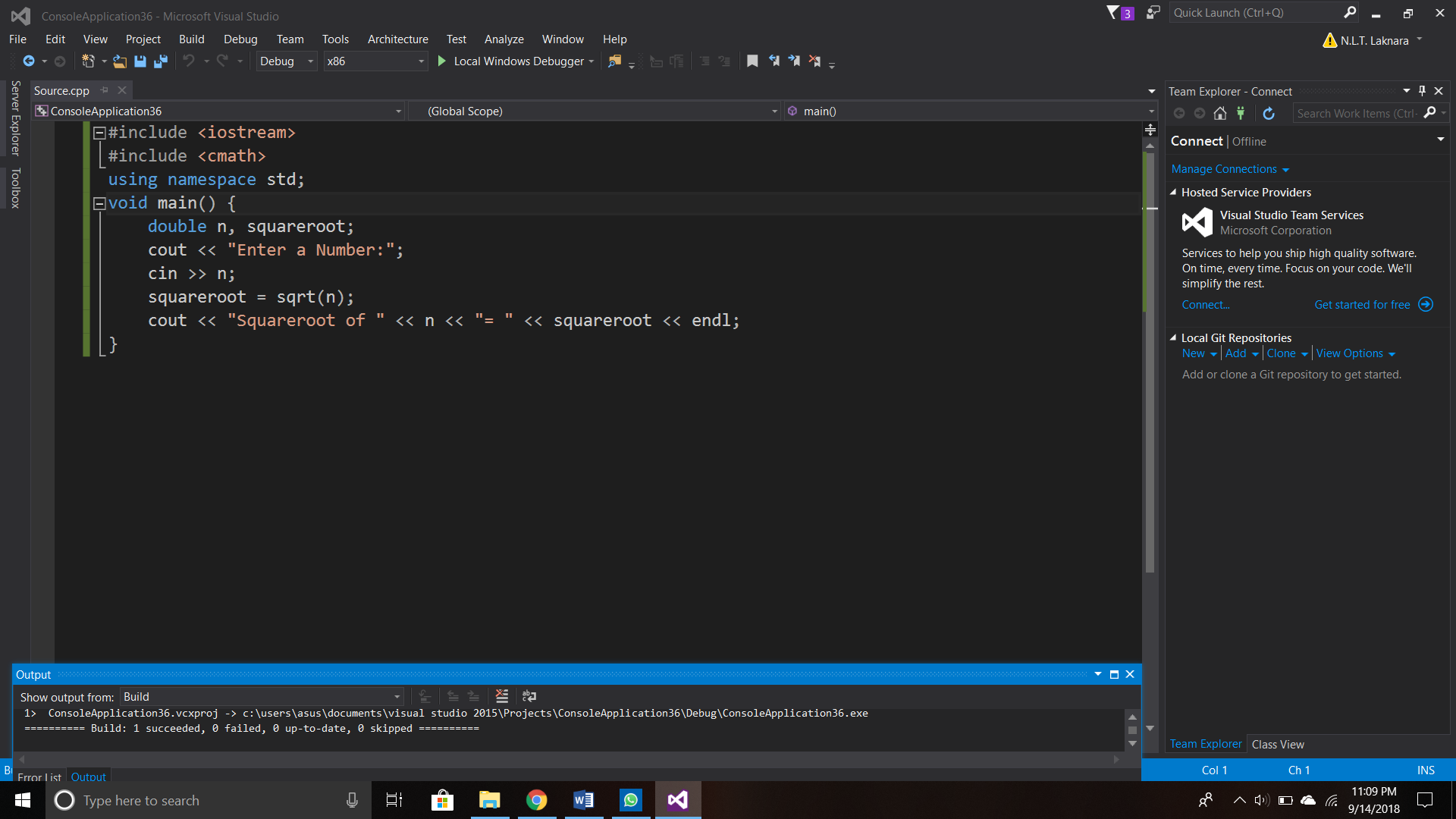
****

Figure 22

#### User-defined Functions

C++ allows programmer to define their own function. A user-defined function groups code to execute a specific task and that group of code is given an identifier. When the function is called from any part of the program, all the codes will be executed which is in the body of function.

There are two possibilities in user-defined functions;

1. User-defined functions which doesn’t have a return value (void)
2. User-defined functions which returns a value

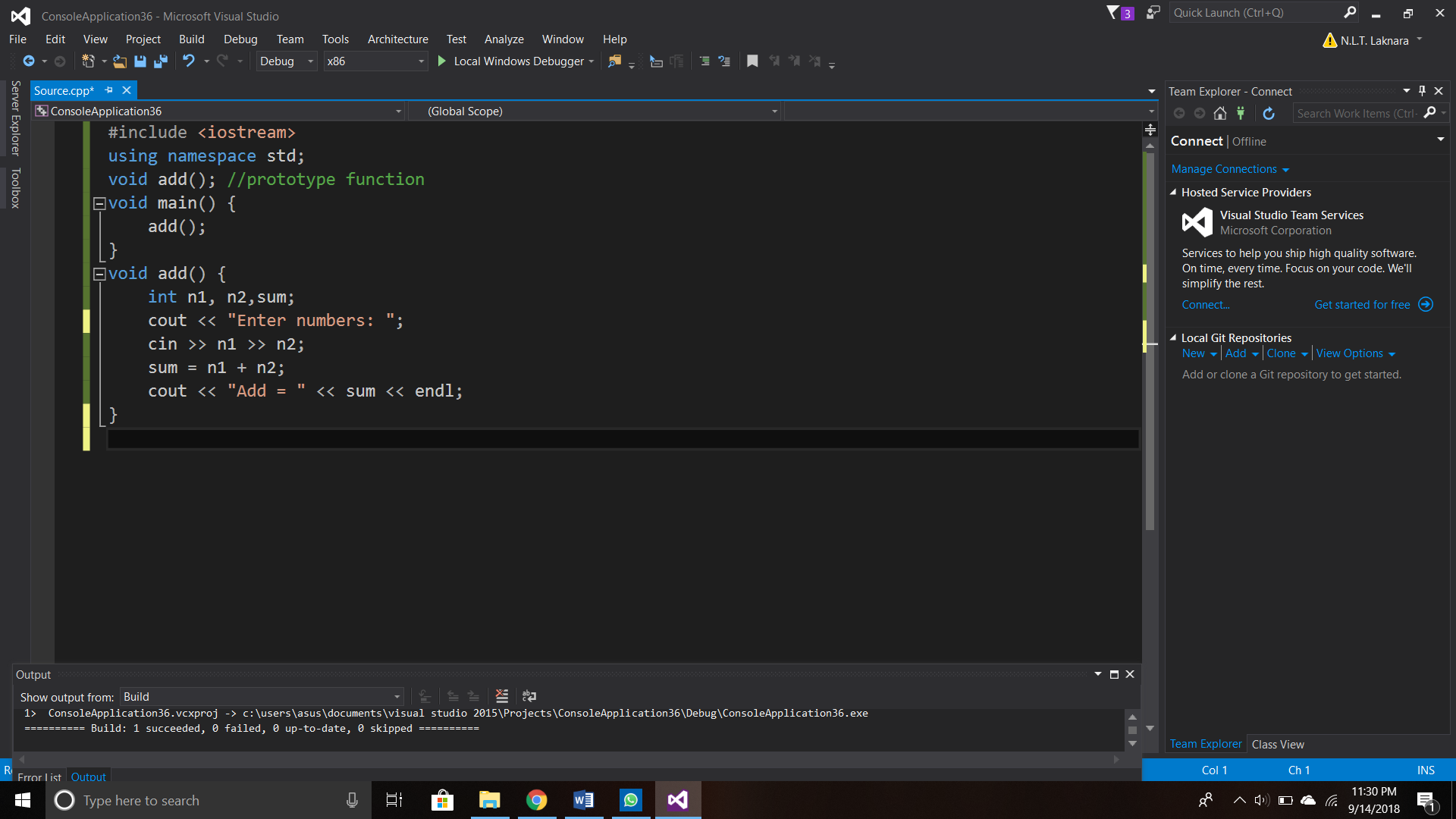


Figure 23

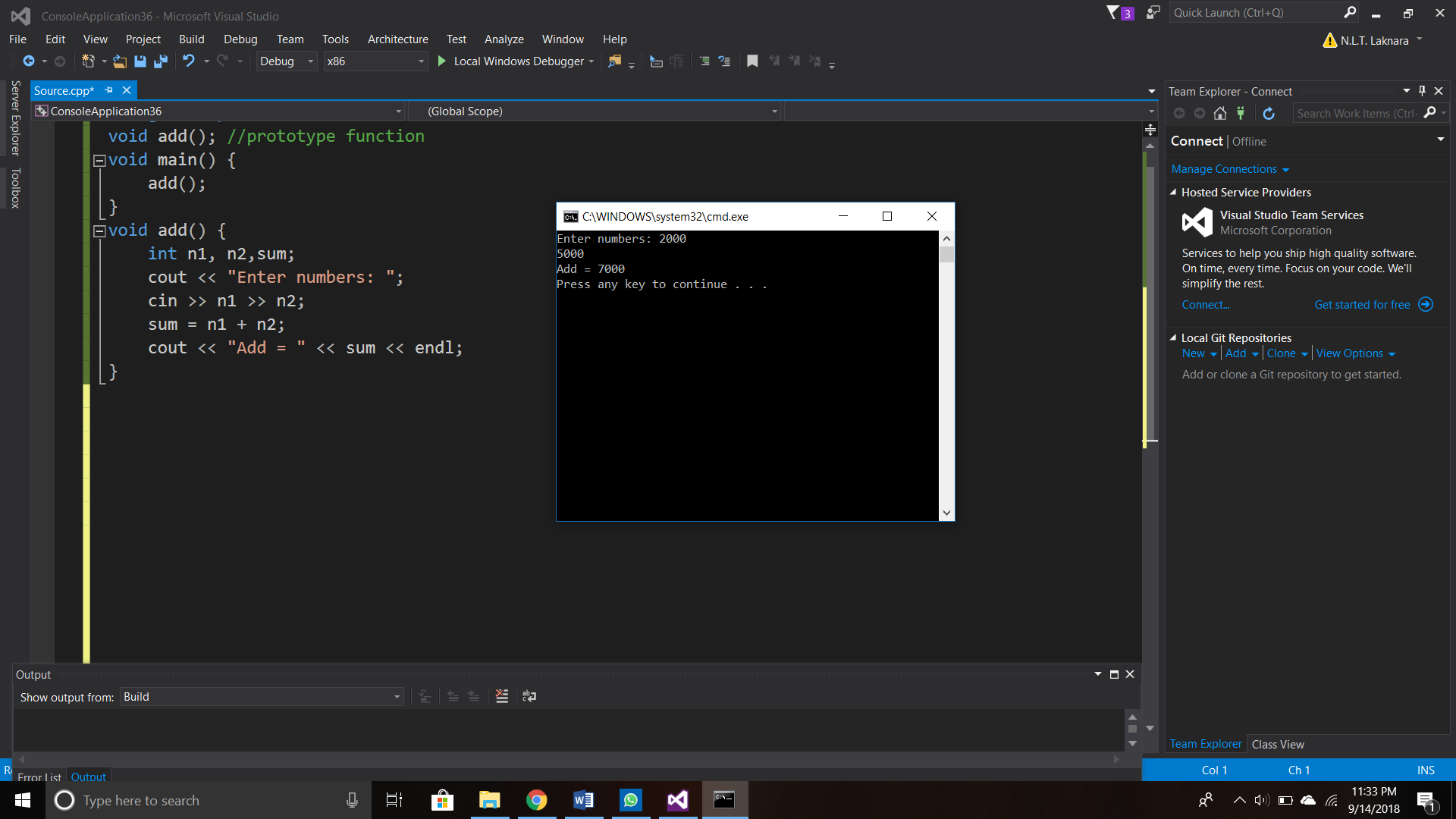


Figure 24

# Task 2 – System Design (LO2)

## 2.1 SRS for Chrystal Clinic Billing System

SRS stands for Software Requirement Specification. SRS document will be specified software, hardware, functional and network requirements of the BestMeal Billing System.

### 2.1.1 Introduction of Chrystal Skin Clinic

BestMeal restaurant is the most popular restaurants in sri lanka. This restaurant has three main meal categories which include; dinner, breakfast and lunch. This restuarannt has excellent experience in making good and healthy foods which gives proficient service to the needs of customers of all ethnicities. It has taken a step forward to provide a worth service to its customers by automating their billing procedure. The system will provide faster and reliable service by distributing correct information resource the customer.

### 2.1.2 Purpose

BestMeal restaurant has taken a decision to provide a valuable service to its customers through an automated online billing procedure. The purpose for this automated online system is to provide a faster and reliable service. Further purposes for the automated online billing system,

* Customers, no need to stay in the queue to pay bills, because through the online billing system they can find out the package name, price, and the full amount.
* Customers can easily select one or more packages without any difficulty via online billing system.
* Company can reduce extravagance for paper works, because the package description, package price and other information will provide through the online billing system.
* Than the manual system the customers will be comfortable to manage their needs as they are being provided a flexible online system authentication.
* All customers’ personal data will be secured with this system; thus the company can provide best solutions for the customer convincingly.

### 2.1.3 Scope of System

In this system user has to insert details for the login purpose. restaurant will manually update the system, hence the customer will update with latest packages, new updated prices and discounts.

User will get to know the available packages and descriptions, which he/ she wants to make an meal order. In this system the user can view the all packages and select what they want to select. The user will be able to select one or more packages and get to know the calculated amount of all the selected meal and the price.

### 2.1.4 Requirements

#### 2.1.4.1 Functional Requirements

Functional requirements define about the software/system functionalities. It is mainly defining how the system is interacting within functionalities. There are functional requirements,

* Log in to the system

User can login to the system anytime by entering correct username and password.

* View available packages

User is allowed to view all packages and decide by observing full description of each meal package.

* Select packages

User can view the all packages available and also user can select one or more packages. User is allowed to select the packages by observing price and the amount.

* Search Packages

User can search packages by entering the package ID.

* Calculate bill

The item/ items which is selected by the user, it will be calculated by the system. It will be shown to the user with in detail window.

#### 2.1.4.2 Non-Functional Requirements

Non-functional requirements define how the system works. There are non-functional requirements can be emphasized.

* Performance

Performance describes how the system/software is response to the given task effectively.

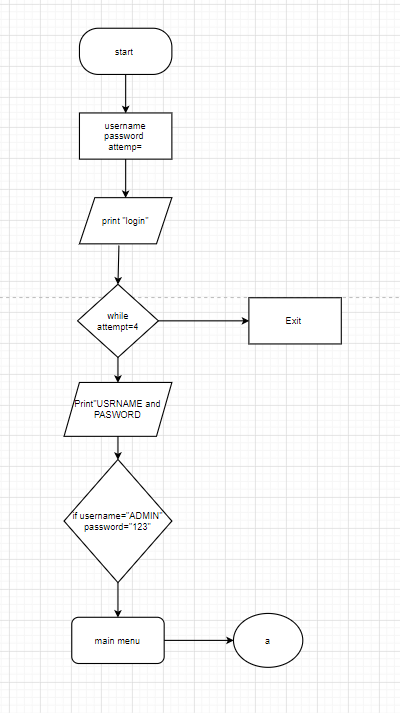
* Reliability

Confidentiality of system data means the reliability. In other hands, reliability can be defined as the less amount of faults and failures. The system must be reliable that it would work for long time effectively.

## 2.2 Flowcharts

A flowchart is a graphical representation of a logic sequence of operations in an information system or program. Program flowchart shows that the sequence of instructions in a program. This emphasizes individual steps and their interconnections. In flowcharts, different symbols are used to draw meaningfully.

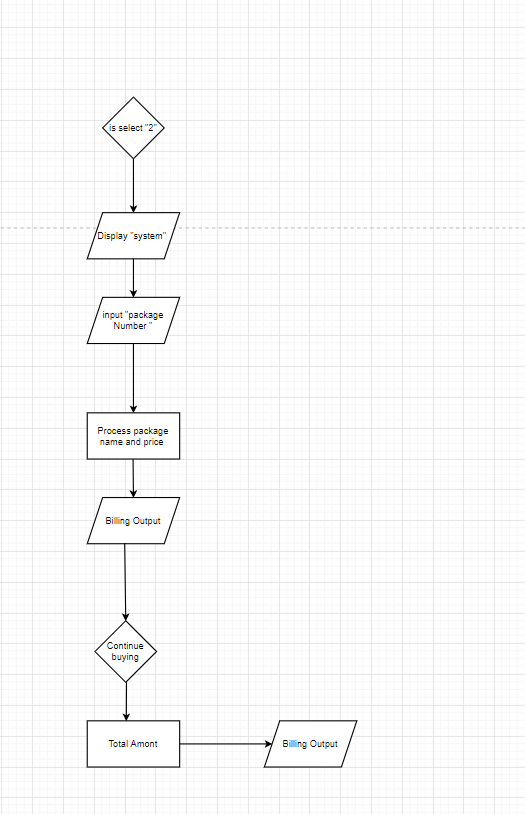
#### Flowchart for Log-in



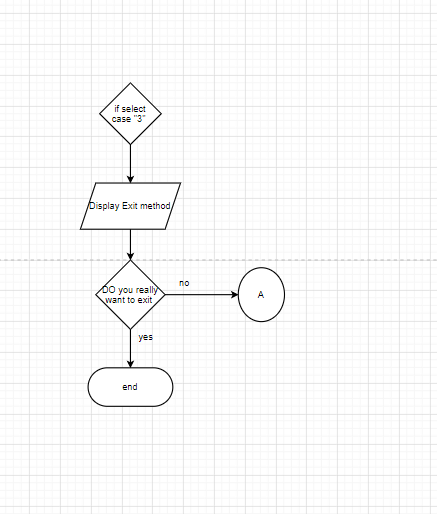
#### .Flowchart for View Available Packages



#### 3.Flowchart for Display the Main Menu (Previous Menu)



#### 4.Flowchart for Exit



# Task 3 – System Design (LO2)

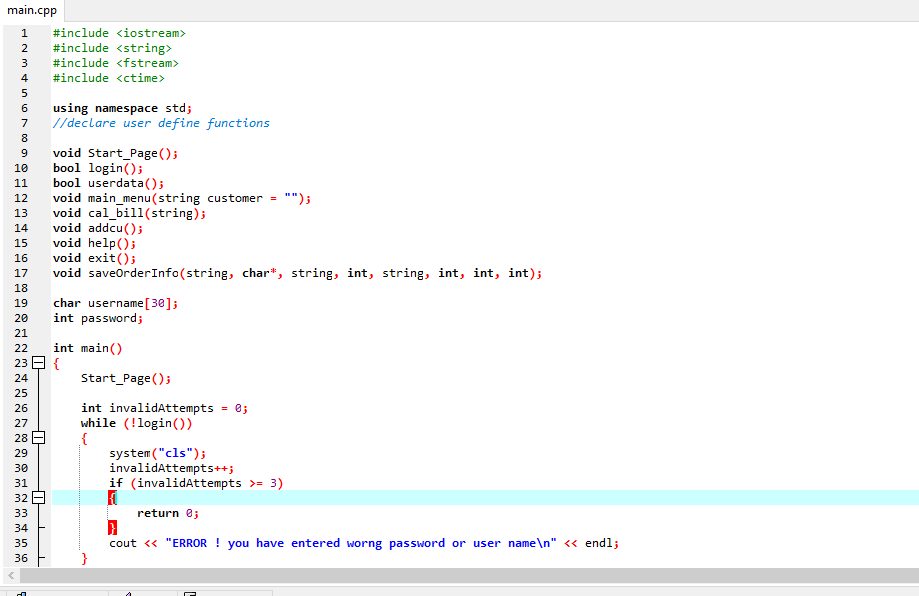


Figure 25coding 1

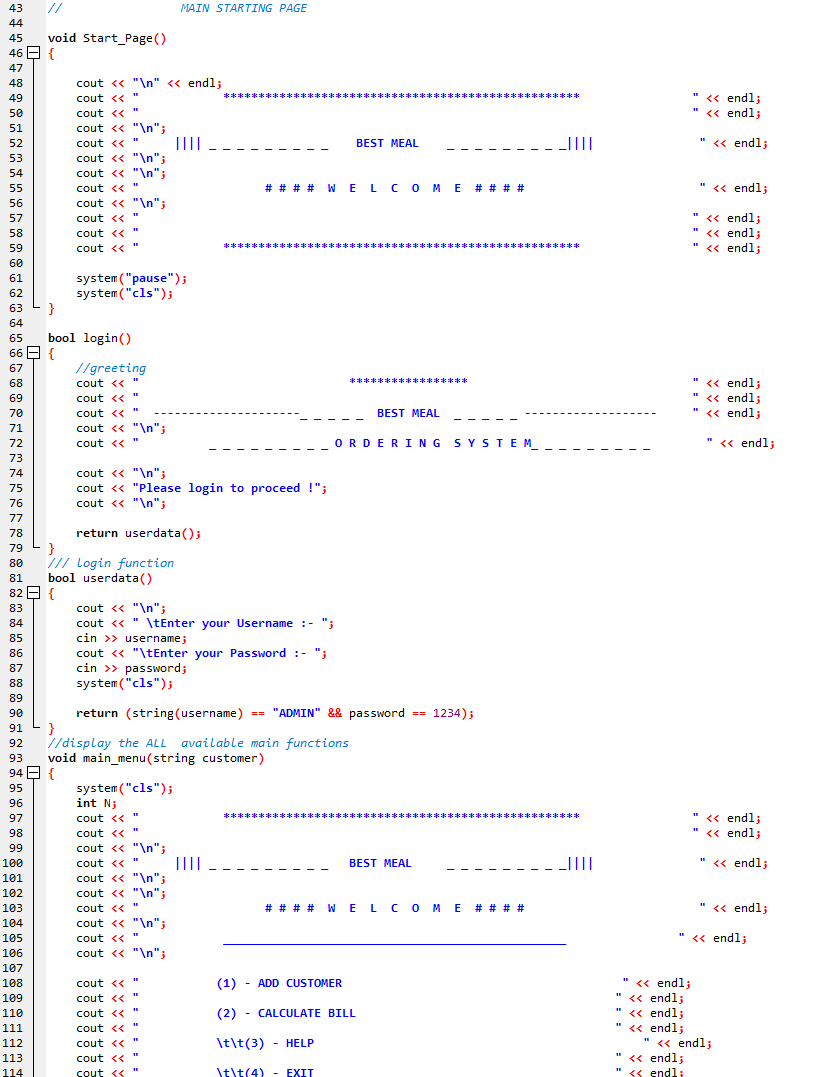


Figure 26coding 2

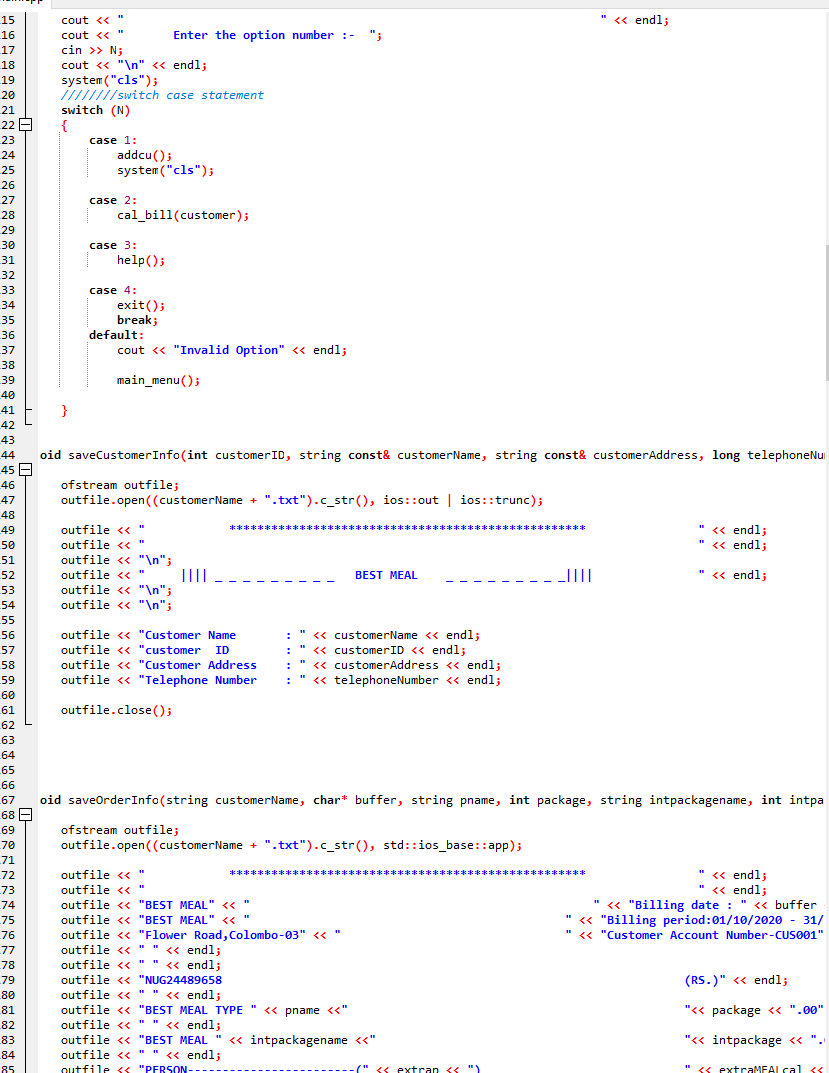


Figure 27coding 3

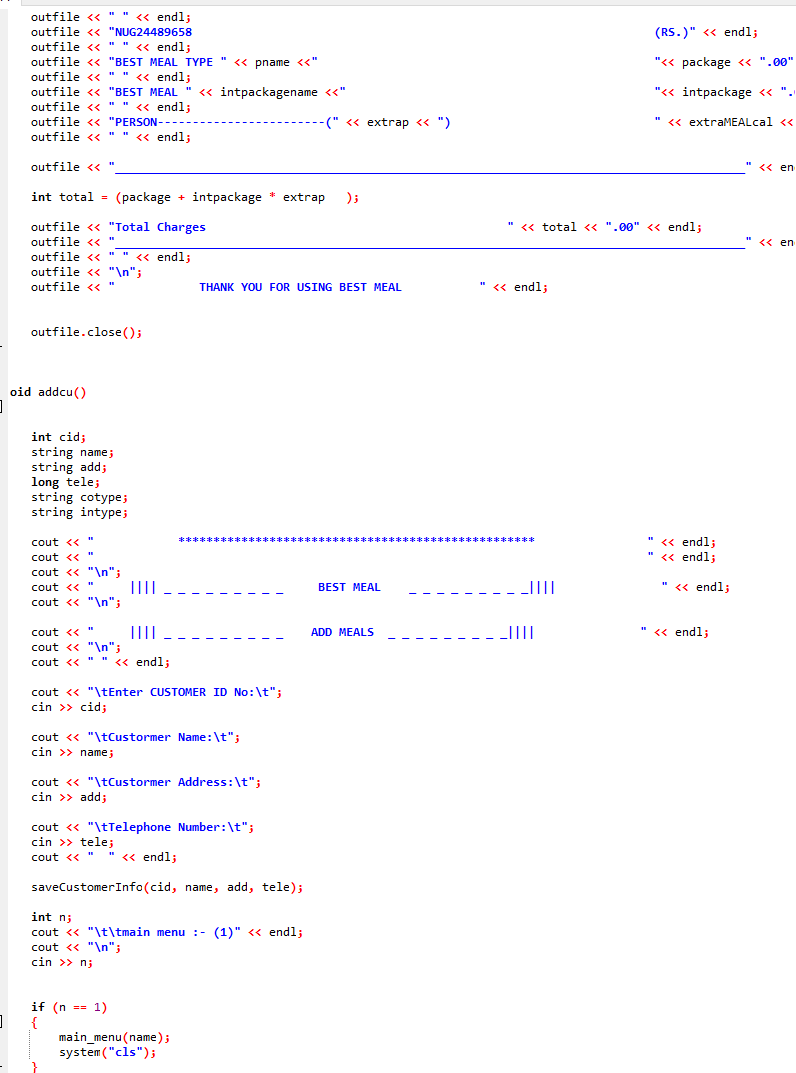


Figure 28coding 4

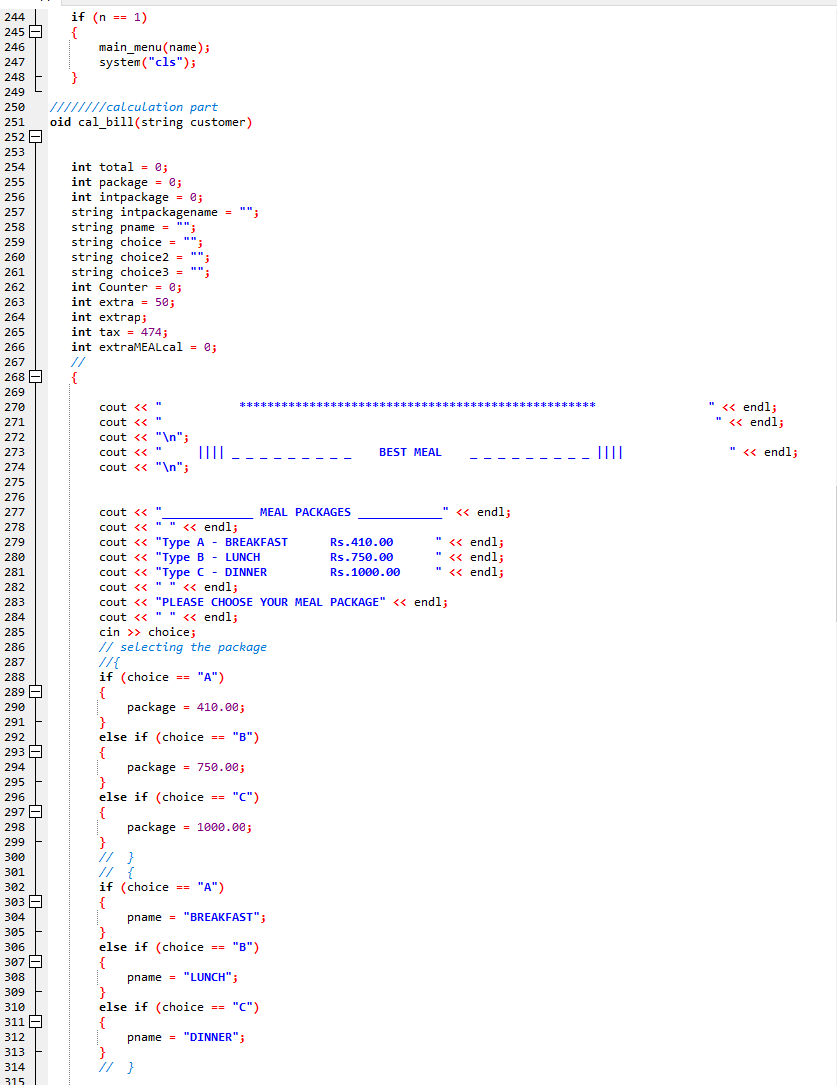


Figure 29coding 4

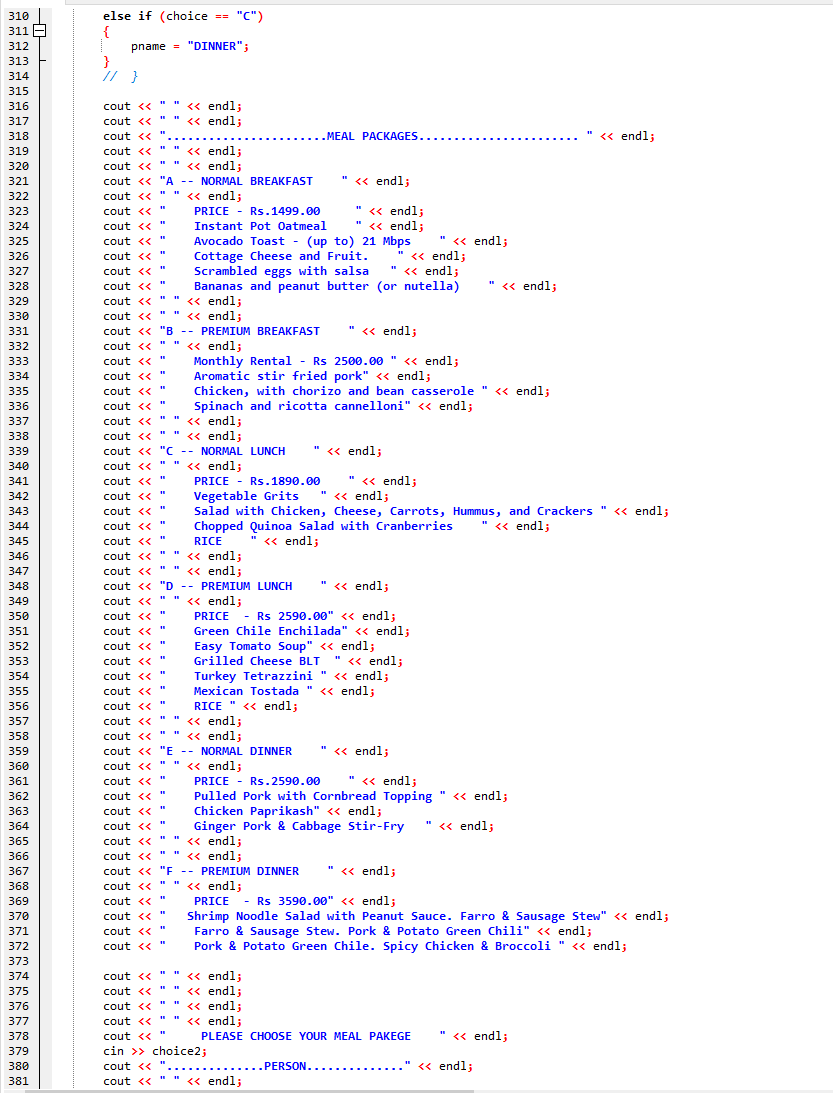


Figure 30coding 5

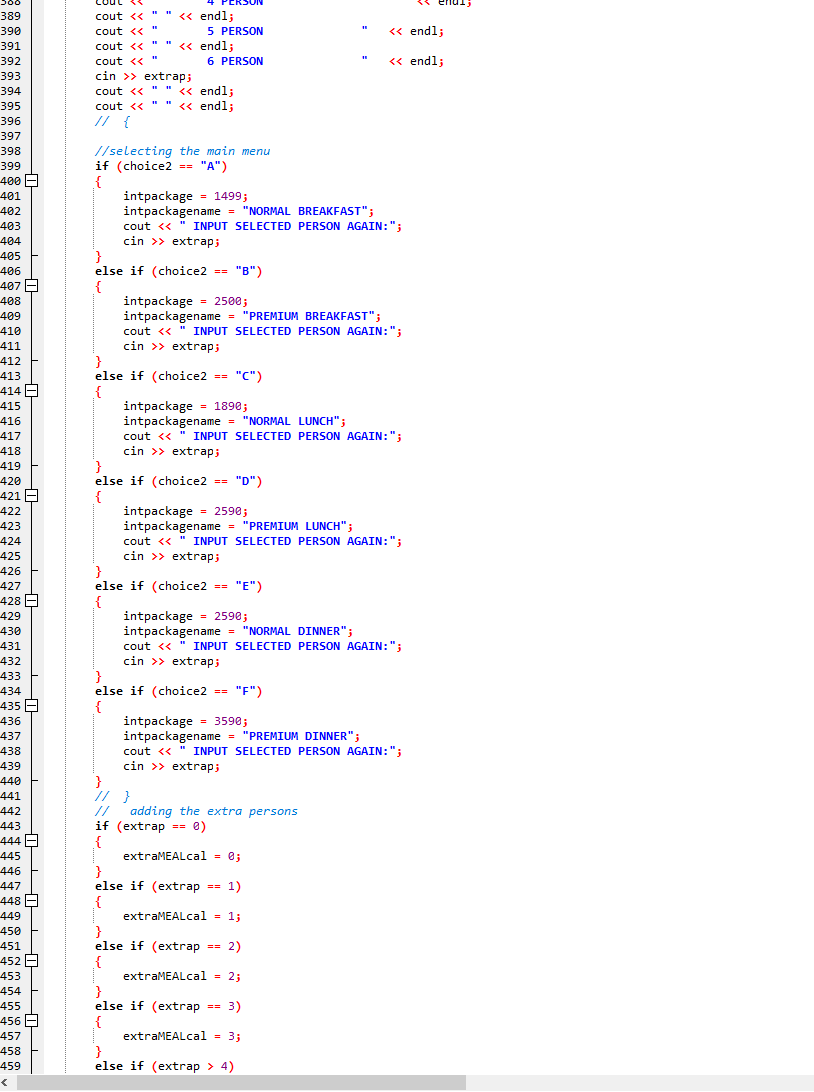
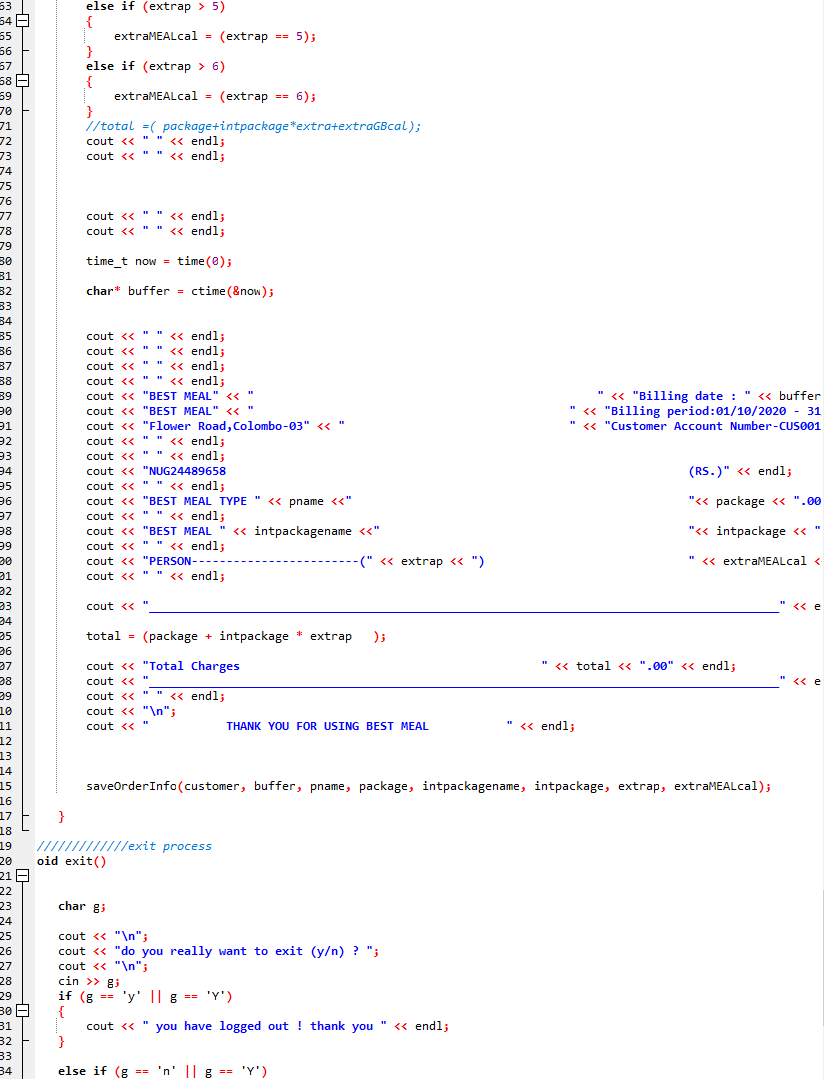
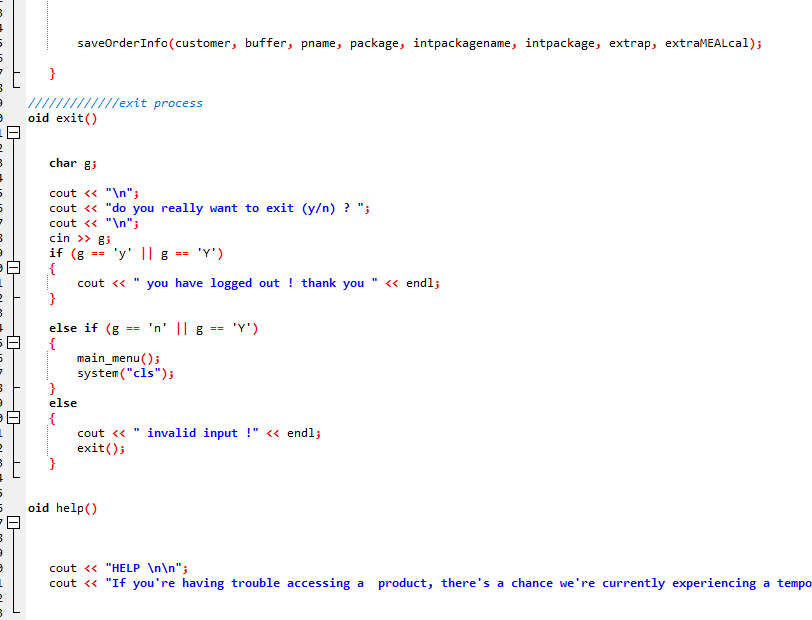


Figure 31coding 6





# Task 4 – Software Testing and Documentation

## 4.1 Introduction to Software Testing

Testing is used to evaluate the system with the purpose of finding whether it meets the specified requirements or not. Testing is the process which executes the system in order to identify the errors, gaps, or missing requirements in opposite to the real requirements.

(Anon., n.d.)

## 4.2 Software Testing Methodologies

Software testing methodologies are the different ways of verifying that the software application is fully tested. There is two methodologies of software testing,

* Functional testing
* Non-functional testing

(Anon., 2021)

### 4.2.1 Functional Testing

It is part of a testing methodology is usually broken down into four components. Those are executed in this order.

* Unit testing

System or software will be implemented according to the modularization. One modular program will be implemented by individually or by team. After implemented the part it will be tested by themselves to make sure whether it’s working in the correct way. Even though it is complex, system will be tested part by part. These tests are usually written by the developers of the module and in a test-driven-development methodology (such as agile, scrum) they’re actually written before the module is created as part of the specification.

* Integration testing

This is the testing of the different modules/ components that have been successfully unit tested when integrated together to perform specific tasks and activities.

* System testing

This testing involves for testing the whole system for error and bugs. This test brings out by interfacing the hardware and software components of the whole system. This testing is listed under the black-box testing method, which the software is checked according to user expectations as well as potential exception and edge condition.

* Acceptance testing

This is the final phase of the functional testing. This involves to make sure that all the product/ project requirements have been met and the end-users and consumer have tested the system to make sure it operates as expected.

(Anon., 2018)

### 4.2.2 Non-Functional Testing

The functional Testing is done using the functional Specifications provided by the client or by using the design specifications like use cases provided by the design team. Perhaps non-functional testing involves testing the application against the non-functional requirements, which typically involve measuring/testing the application against defines technical qualities.

* Performance/load/Stress Testing

There are 2 types of Performance Testing, for ex; the system measuring now a system behaves under increasing load, (Both numbers of users & data volumes), Load Testing is Verifying that the system can operate at the Required response times when subjected to its expected load, and stress testing is finding point (S) in the system.

* Security/ Vulnerability Testing

Security Testing tests the Software for confidentially, integrity, Authentication, Availability, and Non-Reputation, Individual Tests are conducted to prevent any unauthorized access to the Software code.

* Compatibility Testing

This is the Testing that the product or application is compatibility with all the Specific Operating systems, Hardware platforms, web browsers, Mobile devices, & other designed third-party Programs. This tests check that the product works as expected across all the Different Hardware/Software combinations.

* Usability Testing

In this testing The End-User usability aspect of the software the case with which a user can access the Product forms the main testing point, Usability Testing looks at the aspect of testing, learnability, Efficiently, Satisfaction, memorability, & errors.

(Anon., 2018)

### 4.2.3 Black Box Testing

Figure 32

Black Box Texting is also known as behavioral Testing, is a Software Testing method in which the internal Structure/ design/ Implementation the item being Tested not known to the tester. These can be Functional or non-functional, through usually Functional.

**There method attempts to find errors in the following categories;**

1. Incorrect or missing function
2. Interface errors
3. Errors in data structure or external database access.
4. Behavior or performance errors
5. Initialization & Termination errors

### 4.2.4 White Box Testing

This testing also known as clear testing, open box testing, glass box testing. This is the testing method which the structure/ design/ implementation of the product being tested is known to the tester.

## Test Plan

# References

# References

Anon., 2000. *cplusplus.com.* [Online]   
Available at: http://www.cplusplus.com/doc/tutorial/control/  
[Accessed 2018].

Anon., 2011. *TechTerms.* [Online]   
Available at: https://techterms.com/definition/programming\_language  
[Accessed 2021].

Anon., 2012. *BeginnersBook.com.* [Online]   
Available at: https://beginnersbook.com/2017/08/cpp-for-loop/  
[Accessed 2021].

Anon., 2017. *CODINGEEK.* [Online]   
Available at: https://www.codingeek.com/tutorials/c-programming/string-library-functions-explanation-with-example/  
[Accessed 2018].

Anon., 2018. *inflectra.* [Online]   
Available at: https://www.inflectra.com/ideas/topic/testing-methodologies.aspx  
[Accessed 2021].

Anon., 2018. *PadaKuu.* [Online]   
Available at: http://www.padakuu.com/article/21-scope-and-classification-of-metrics  
[Accessed 11 September 2018].

Anon., 2019. *Programiz.* [Online]   
Available at: https://www.programiz.com/cpp-programming/if-else  
[Accessed 1 january 2021].

Anon., 2021. *inflectra.* [Online]   
Available at: https://www.inflectra.com/ideas/topic/testing-methodologies.aspx  
[Accessed 2021].

Anon., n.d. *Fcu.edu.* [Online]   
Available at: https://www.cs.fsu.edu/~myers/c++/notes/control2.html

Anon., n.d. *PadaKuu.* [Online]   
Available at: http://www.padakuu.com/article/1-difference-between-manual-and-automated-system-manual-system-vs-automated-system  
[Accessed 10 September 2018].

Anon., n.d. *Programiz.* [Online]   
Available at: https://www.programiz.com/cpp-programming/if-else

Anon., n.d. *Programiz.* [Online]   
Available at: https://www.programiz.com/csharp-programming/ternary-operator

Anon., n.d. *Programiz.* [Online]   
Available at: https://www.programiz.com/cpp-programming/do-while-loop

Anon., n.d. *Programiz.* [Online]   
Available at: https://www.programiz.com/cpp-programming/library-function  
[Accessed 2021].

Anon., n.d. *T4tutorials.* [Online]   
Available at: https://t4tutorials.com/non-functional-requirementsnfrs-software-engineering-se/  
[Accessed 2018].

Anon., n.d. *Techopedia.* [Online]   
Available at: https://www.techopedia.com/definition/25972/modular-programming  
[Accessed 2021].

Anon., n.d. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/index.htm