Assignment Cover Sheet

Qualification	**Module Number and Title**
Higher Diploma in Computing and Software Engineering	CSE 4002 Fundamentals in Programming
Student Name & No.	**Assessor**
M.A. Chamath Shyamal CL/HDCSE/95/43	Deshan Bulathsinghala
Hand over date	**Submission Date**
19.02.2021	22.04.2021

Assessment type	**Duration/Length of Assessment Type**	**Weighting of Assessment**
Coursework	Report and Software Submission (3000 words)	100%

Learner declaration

I, M.A. Chamath Shyamal/CL/HDCSE/95/43, 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	**: Deshan Bulathsinghala**
Assignment	**:** FP

Strong features of your work:

Areas for improvement:

Marks Awarded:

Acknowledgement

Primarily I thank god for being able to complete this assignment in a successful manner. Thus, I take this opportunity to express my deep sense of gratitude and my profound respect to the lecturer who guided and inspired me in doing this assignment. I had to get the guidance of a respected and responsible person at the preparation time and while continuing the project. So, I would like to thank our lecturer Mr. Deshan Bulathsinhala whose valuable guidelines and consultations been the ones that helped me patch this assignment and make this full proof success and finalizing this successfully.

And also, his instructions which were given underlying the structure has served as the major contributor towards in completing this assignment and his instructions about the programming and coding was more help in implementing this billing application. Through those and by this golden opportunity, I got the full knowledge on the Fundamentals in Programming. I hope this knowledge you gave me will help me in learning computer languages as well as my career. I really thankful for you because of the massive courage you had to teach us.

Then I would like to thank Mr. Chathura, Miss Manoda and Miss Ishani who is with us and help us in many sides since the beginning of the Bridging Program. However finally, I am really grateful because I managed to complete this assignment within the time period given by our lecturer Mr. Deshan. Thank you all!

Content

Introduction

This Assignment is regarding the Fundamentals in Computer Programming Module and this is the fifth assignment we got in our HD Program. Basically, this coursework covers structured programming concepts, designing of a basic structured computer program, developing of a modularized computer programme for a prepared design, software testing and documentation. Studying about Fundamentals in computer programming is more important to a student who hopes to become a Software Engineer as Fundamentals in computer programming directly explains the basics in building software and how complex codes are made using the basic stuffs. In my assignment, I have ideally shown some theory explanations at the beginning of the document.

The primary and main objective of implementing this is creating a billing application for Rathnayaka Gyms which is hoping to automate their manually using billing system. All of the healthcare services they provided are being delivered by a team of well trained, committed and passionate professionals, whilst being managed and guided by some of the most qualified and respected experts of the healthcare and fitness industry. Further they sell food supplements and fitness equipment for both members of the gym and customers who come from outside. They have a strong chain of branches which located in Kaduwela, Kottawa, Malabe, and Avissawella. Thus, they hope to make it easier their billing process by a automated application.

This document includes explanations about both structured oriented and object-oriented programming methodologies, various types of selections in computer programming, repetitions used in programming, modularization, functions, a system requirement specification document which deeply explains about the applications which is implemented, flowcharts, pseudocodes, testing technique used, a test plan and test cases which are needed to implement a system in an effective and user-friendly manner. I have used C++ programming language in building this billing application. The main purposes of this billing application are to view available fitness package details, view available food supplements, view fitness equipment and to calculate bill.

Hope this assignment will be a clear and ideal one.

Task 01 (Programming methodology)

1.1) Programming Methodology

Computer Programming can be defined as a set of instructions which are used for creating, modifying and developing computer's functionalities. Simply, it says programming is an intermediate step between specifying and implementing a programming system. During that step, designing of program architecture, abstractions and coding them into a programming language are done. Programming can be considered in two ways as technology (practical techniques, current tools, standards) and science (fundamental concepts). Thus, Computer Programming is some sort of a complex procedure which can be done by both programming specialist and non-specialist. In here, I mentioned as non-specialist for like consumers who are able to change the settings of their cellular phone. The instructions which are used in programming can be written using a particular programming language. By now, there are many programming languages with unique methods which are also called syntaxes in the software industry in order to make computer programs for solving real life problems such like payroll processing, examination result processing, etc.

Two Types of Programming Methodologies

There are many programming methodologies such as functional programming, procedural programming, object-oriented programming, logical programming and so on. In here, I'm going to highlight only two programming methodologies. They are procedural programming and object-oriented programming.

1.1.1- Structured Oriented Programming Methodology

This is also called as Procedural Programming. This methodology was invented in 1966 by Corrado Böhm and Giuseppe Jacopini, who exemplified theoretical computer program design using sequences, loops and decisions. Then Edsger W.Dijkstra developed this methodology as a mostly used method around 1960s. And also, this is the oldest methodology in the software industry. Structured programming makes the program easier to understand and modify and this methodology uses a top-down design approach in which a program is divided into multiple subsystems/blocks of code or functions in order to minimize the complexity. Only after taking all those subsystems, the whole system is designed. SOP methodology is well going with small, not complex projects. Languages like C, COBOL, Pascal, etc. are used in structured oriented programming.

There is an ideal and clear example below which depicts the concept behind Structured Oriented Programming.

"For a calculator program that does addition, subtraction, multiplication, division, square root and comparison, each of these operations can be developed as separate procedures. In the main program each procedure would be invoked on the basis of user's choice." (Anon., 2021) (Dale Janssen, 2008)

1.1.2 - Object Oriented Programming Method

This was introduced by 1970s – 1980s. This programming methodology became a dominant programming methodology right after Structured Oriented Programming because this Object-Oriented Programming came up with more advantages than Structured Oriented Programming. Few reasons for this methodology to become popular are it well goes with client imaginations and intentions, uses a bottom-up design approach in which individual parts of the system are specified deeply and linked until a complete system is built, can start implementing without wasting the time by waiting until all the requirements are being collected, client can involve and make modifications while building of the program and so on. So those reasons highlight this methodology is based on the objects/entities. Some distinguished features of OOP are Class, Objects, Inheritance, Encapsulation and Polymorphism. Simply in here, program is divided into small parts called objects. Therefore, a computer program is organized as collection of objects. Languages like C++, C#, Java, Python, etc. are used in object-oriented programming.

There is an ideal and clear example below which depicts the concept behind Object Oriented Programming

"If we have to develop a payroll management system, we will have entities like employees, salary structure, leave rules, etc. around which the solution must be built." (Anon., 2021) (Nygaard, 1986) (Roy, 2004)

1.1.3 - Usages of Basics of Computer Programming

So basically, having a proper idea on what do we commonly found in computer programming is a must to begin a project using a programming language. Because of that, I am going to briefly describe few basic stuffs in addition to Controlling Statements. Therefore, following explanations will be focused on programming environment, basic syntax, data types, variables, keywords, and operators. Then after those explanations, explanations about the controlling statements will be described. (Matt Wilwy, 2020)

- Programming Environment

A Programming Environment means it's just like a platform for programmers which we can do our programming in an effective manner by making it easier programmer's workload. Basically, a simple programming environment can include a text editor used to write computer programs, an assembler or compiler converts assembly language into machine code/binary format, an interpreter to execute the programs directly, source level debugging facilities, input-output drivers having OS as well as a file system. In addition to that type of simple programming environments, there are bit complex programming environments also. Therefore, its needed to download a software in order to write computer programs, compile, and execute those programs. Thus, it is clear that this programming environment term is widely used to define any hardware and software in the environment used by programmers.

- Basic Syntax

For programming languages, syntax is a set of grammar and spelling rules. It uses a character structure that can be used by computers. For example, when a user tries to execute a command without the correct syntax, a syntax error occurs and the program usually fails.

- Data Types

A Data Type defines a particular type of data which can be interpreted using a computer program for the understanding. In simple manner, it says how a programmer is going to use a particular data type in the program. Thus, programming languages allows different sorts of data types such as integer, string, real, character, Boolean and so on. So, it's convenient that data types used in programming can be decimal, alphanumeric, numeric, etc. More importantly, when writing a computer program, specifying the type of data is essential. And also, the reason for mentioning as specifying data types is different from programming language to language is because in C language it uses int to specify integer. But specifying integer in other programming languages may be different.

- Variables

Variables are like containers which hold information. The aim of using variables is to referencing and store values in memory. It's advantageous because those data can be used while continuing the coding of the program. And also, here, data can be labelled giving a descriptive name in order to clearly identify the programs. Further, variable declaration means the programmer telling the computer that programmer needs locations to store values.

- Keywords

A keyword means it can be parameters or instructions in which a particular word is stored and has a distinctive meaning. Different data types are specified using different keywords in different programming languages as they consist with keywords that cannot be used as variable names. Moreover, its unable to use a stored keyword to name a variable.

Few Examples:

- ❖ In C language: struct, float, default, case
- ❖ In Java: class, double, static, switch

- Operators

Operators are basic foundation consists with symbols for beginners to achieve particular mathematical, relational or logical operations and produce final output. In programming, most common and widely used operators are Arithmetic operators, Logical operators, Assignment operators, Relational operators. Moreover, there are operators like Bitwise Operators, Comparison Operators, Error Control Operators, Execution Operators, Incrementing/Decrementing Operators and so on. Few common examples: +, /, !=, >=, &&

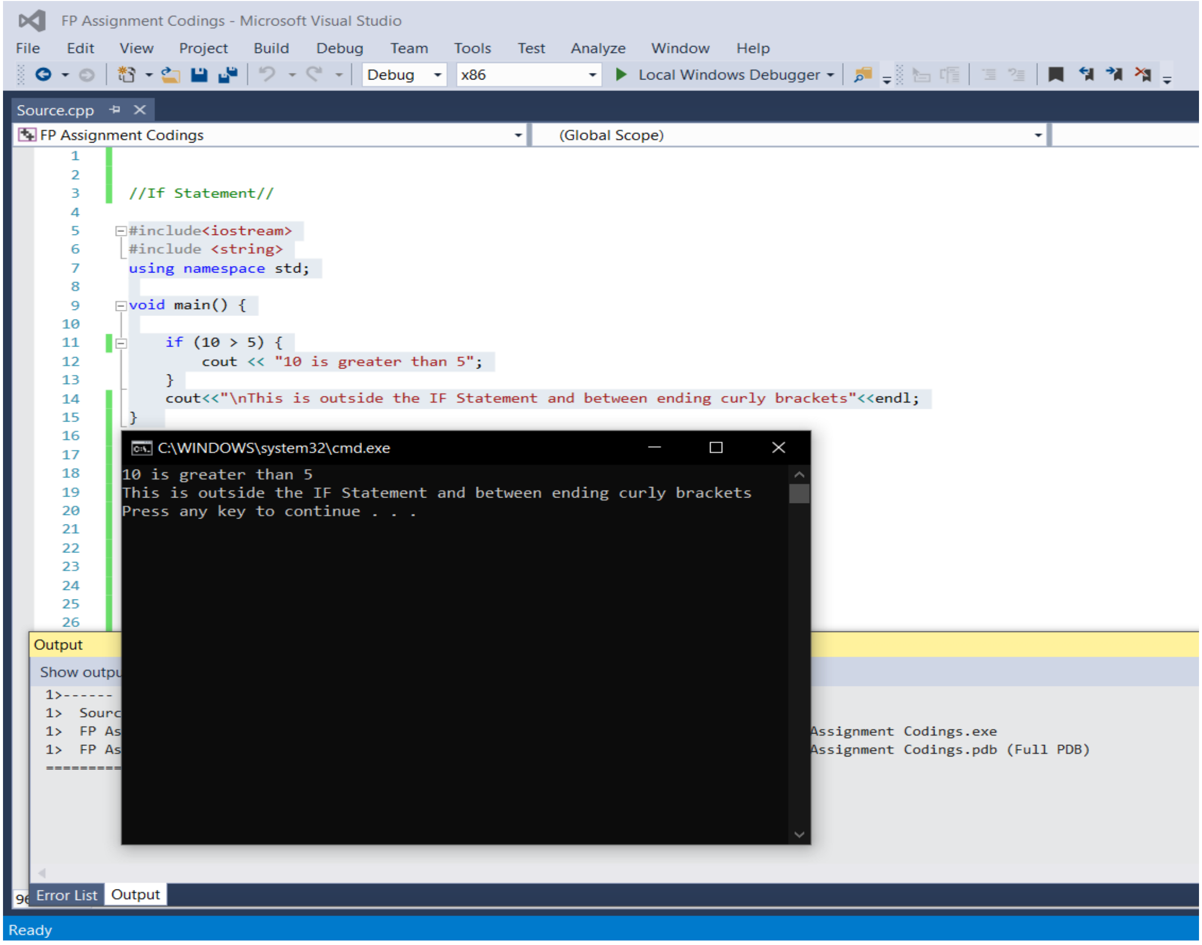
l.2) Controlling statements

In order for a programmer to evaluate or test a decision/controlling structure, the programmer must assign one or more conditions, and one or more statements may have to be executed for the condition to be determined whether it is true and to execute another statement if the condition is determined to be false.

[1.2.1] Selections in programming

1.2.1.1- 'If' Statement

When it comes to if statement, it will return either true or false. The process happens inside the if statement is, if the codes inside the 'if' are true, then the statement will be executed and display information. If not, that means if the codes inside the 'if' are false, then the instructions after the closing curly brackets of the if statement will be executed.



```
//If Statement//

#include<iostream>
#include <string>
using namespace std;

void main() {

    if (10 > 5) {
        cout << "10 is greater than 5";
    }
    cout<<"\nThis is outside the IF Statement and between ending curly brackets"<<endl;
}
```

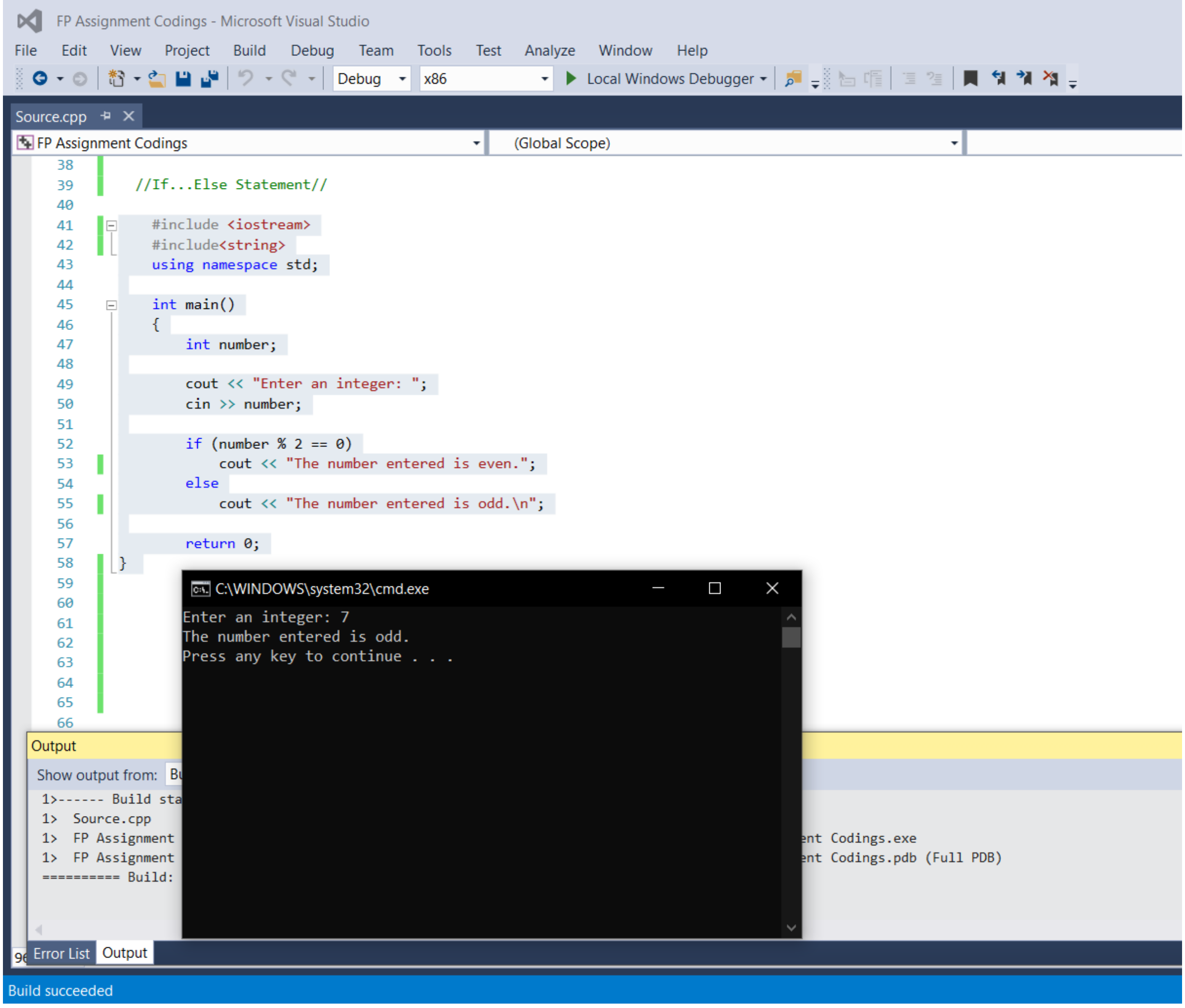
```
10 is greater than 5
This is outside the IF Statement and between ending curly brackets
Press any key to continue . . .
```

```
1>------
1>  Sourc
1>  FP As                                                      Assignment Codings.exe
1>  FP As                                                      Assignment Codings.pdb (Full PDB)
=========
```

As shown in the above screenshot, it should be displayed as `10 is greater than 5`. Through the screenshot of the output, that is proven. In the code, I have put `cout<<"this is outside the IF Statement and between ending curly brackets"<<endl;` after closing curly brackets. So, this (`this is outside the IF Statement and between ending curly brackets`) also has displayed in the output.

1.2.1.2- ‘If…Else’ Statement

If statement can be followed by another statement which is called as else statement. That statement will be executed if the Boolean expression is false. When you have to decide between two options, if…else statement works well and useful. In simple manner, if…else means it executes two different codes depending on true or false.



```
//If...Else Statement//

#include <iostream>
#include<string>
using namespace std;

int main()
{
    int number;

    cout << "Enter an integer: ";
    cin >> number;

    if (number % 2 == 0)
        cout << "The number entered is even.";
    else
        cout << "The number entered is odd.\n";

    return 0;
}
```

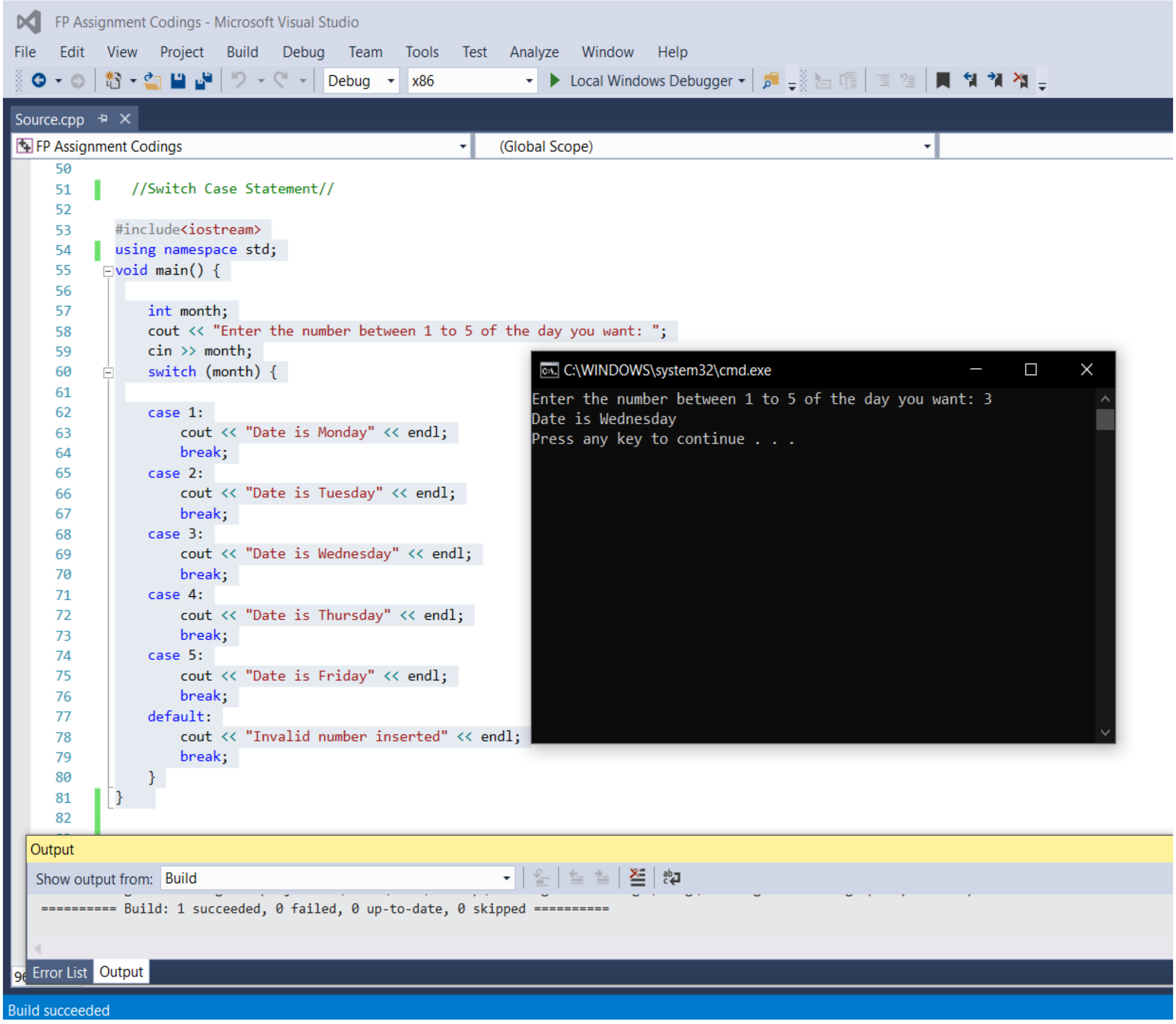
```
Enter an integer: 7
The number entered is odd.
Press any key to continue . . .
```

```
1>------ Build sta
1>  Source.cpp
1>  FP Assignment                                                  ent Codings.exe
1>  FP Assignment                                                  ent Codings.pdb (Full PDB)
========== Build:
```

As shown in the above screenshot of coding, it should be displayed as a particular entered number is odd or even. The statement which is under the else will be executed if only the statement which is under if becomes false. So that is clearly depicted in the above screenshot.

1.2.1.3- ‘Switch- Case’ Statement

The switch statement is faster than if…elseif…else statement. The switch statement is a statement that allows you to check the compatibility of a variable with a set of values. Each value is called a case. It is better to use switch case statement to check on the value of a single variable. In here, an integer value should be provided for the expression. After matching an expression value and a case value, body of that matching expression will be executed. At last, the switch is terminated using break statement. The default case body will be executed if any case doesn’t match.


FP Assignment Codings - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp
FP Assignment Codings
(Global Scope)
C:\WINDOWS\system32\cmd.exe
Output
Show output from: Build
Error List
Output
Build succeeded

```
//Switch Case Statement//

#include<iostream>
using namespace std;
void main() {

    int month;
    cout << "Enter the number between 1 to 5 of the day you want: ";
    cin >> month;
    switch (month) {

    case 1:
        cout << "Date is Monday" << endl;
        break;
    case 2:
        cout << "Date is Tuesday" << endl;
        break;
    case 3:
        cout << "Date is Wednesday" << endl;
        break;
    case 4:
        cout << "Date is Thursday" << endl;
        break;
    case 5:
        cout << "Date is Friday" << endl;
        break;
    default:
        cout << "Invalid number inserted" << endl;
        break;
    }
}
```

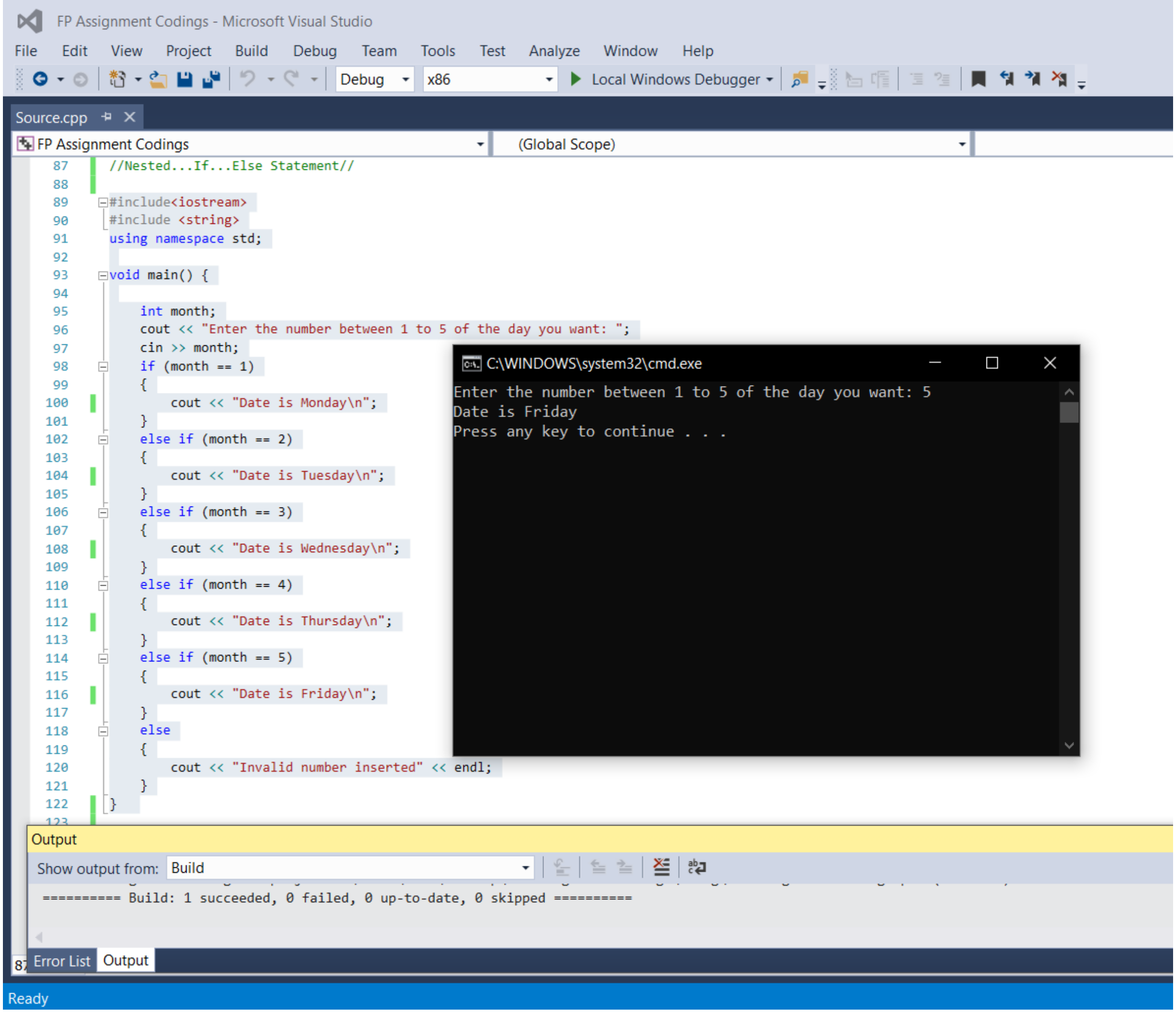
```
Enter the number between 1 to 5 of the day you want: 3
Date is Wednesday
Press any key to continue . . .
```

```
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

According to the example I have chosen to explain the Switch case statement, if a user inserts number 3 as shown in the screenshot, then only the case which is assigned for number 3 will has executed. Moreover, if a user entered a number which is not between 1 and 5, then the statement inside the default (`Invalid number inserted`) will be executed.

1.2.1.4- ‘Nested If… Else’ Statement

Nested if else means an if…else statement includes in the body of a particular if or else statement. In Nested If Else, the expressions which are testing are assesses from top to bottom. The accompanying statements are executed whenever the actual test expression occurs. If all tested expressions been false, then the default statement will valid and it will be executed. Therefore, using this Nested If Else, it can be checked multiple test expressions and also can execute various codes for more than two conditions. Simply, to check many conditions, nested if… else statement is very useful in programming.


FP Assignment Codings - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp
FP Assignment Codings
(Global Scope)
C:\WINDOWS\system32\cmd.exe
Output
Show output from: Build
Error List
Output
Ready

```cpp
//Nested...If...Else Statement//

#include<iostream>
#include <string>
using namespace std;

void main() {

    int month;
    cout << "Enter the number between 1 to 5 of the day you want: ";
    cin >> month;
    if (month == 1)
    {
        cout << "Date is Monday\n";
    }
    else if (month == 2)
    {
        cout << "Date is Tuesday\n";
    }
    else if (month == 3)
    {
        cout << "Date is Wednesday\n";
    }
    else if (month == 4)
    {
        cout << "Date is Thursday\n";
    }
    else if (month == 5)
    {
        cout << "Date is Friday\n";
    }
    else
    {
        cout << "Invalid number inserted" << endl;
    }
}
```

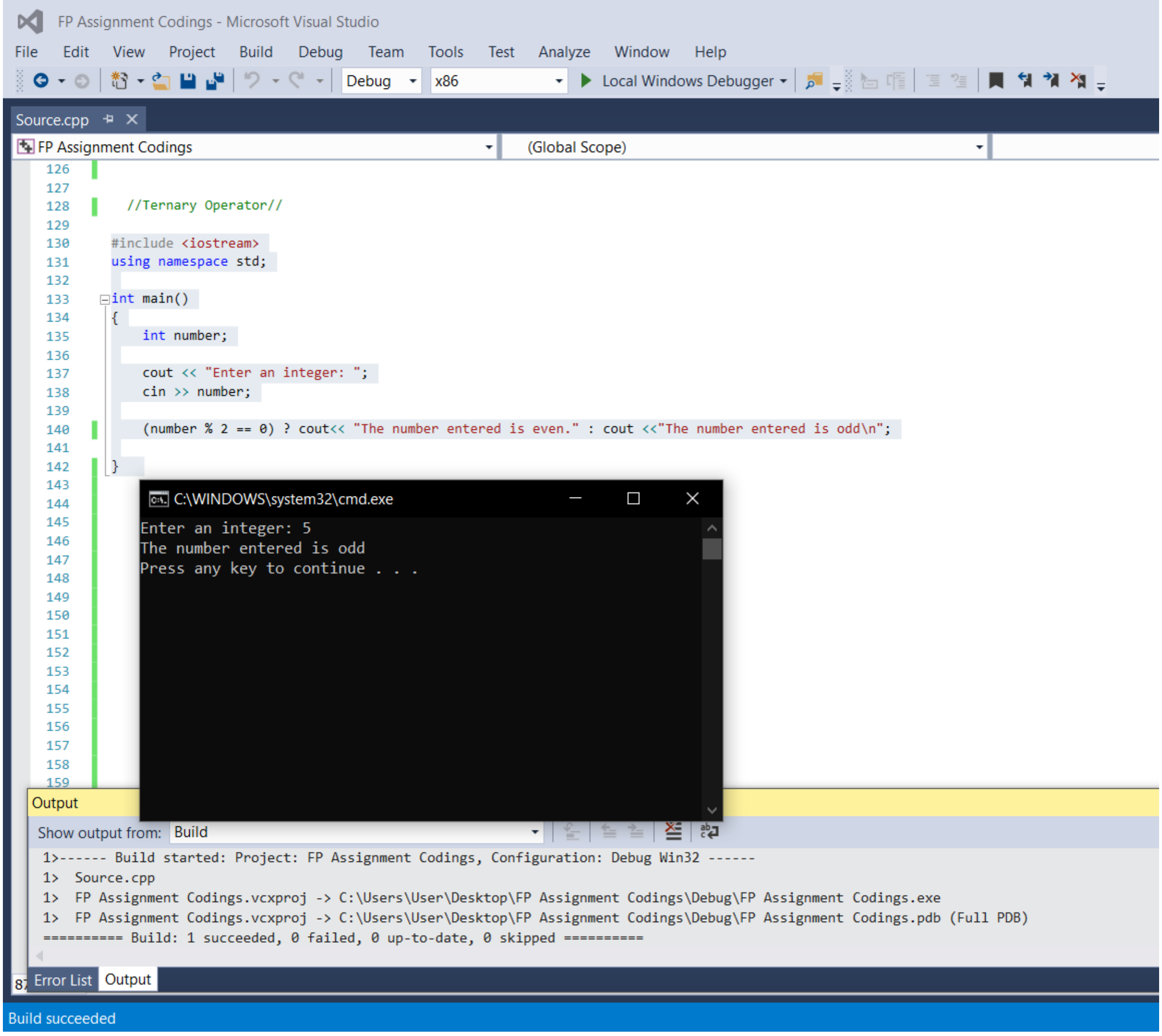
```
Enter the number between 1 to 5 of the day you want: 5
Date is Friday
Press any key to continue . . .
```

```
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

As shown in the above screenshot of coding, there can see I have used many else…if commands. So that’s the advantage in here. When the statement inside the if is false, next else..if statement will be checked, if that statement is also false, then then next else…if statement will be executed. Like that until it meets the user inserted input, it will pass by each command. If it did not find the correct statement related to the user’s input. It will display "Invalid number inserted".

1.2.1.5- ‘Conditional / Ternary Operator

This is bit similar to if…else statement. But due to the high performance of the program, Ternary operator is more useful than if…else statement. So, it’s clear that in here, the if…else statements are shortened in an effective manner. This operator can be used in some programming languages that use three operands instead of the one or two as most operators use.


FP Assignment Codings - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger
Source.cpp
FP Assignment Codings
(Global Scope)
C:\WINDOWS\system32\cmd.exe
Output
Show output from: Build
Error List Output
Build succeeded

```
//Ternary Operator//

#include <iostream>
using namespace std;

int main()
{
    int number;

    cout << "Enter an integer: ";
    cin >> number;

    (number % 2 == 0) ? cout<< "The number entered is even." : cout <<"The number entered is odd\n";

}
```

```
Enter an integer: 5
The number entered is odd
Press any key to continue . . .
```

```
1>------ Build started: Project: FP Assignment Codings, Configuration: Debug Win32 ------
1>  Source.cpp
1>  FP Assignment Codings.vcxproj -> C:\Users\User\Desktop\FP Assignment Codings\Debug\FP Assignment Codings.exe
1>  FP Assignment Codings.vcxproj -> C:\Users\User\Desktop\FP Assignment Codings\Debug\FP Assignment Codings.pdb (Full PDB)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Unlike in If…else, though conditional operator the output is satisfied well with one coding line.

So as in the screenshot;

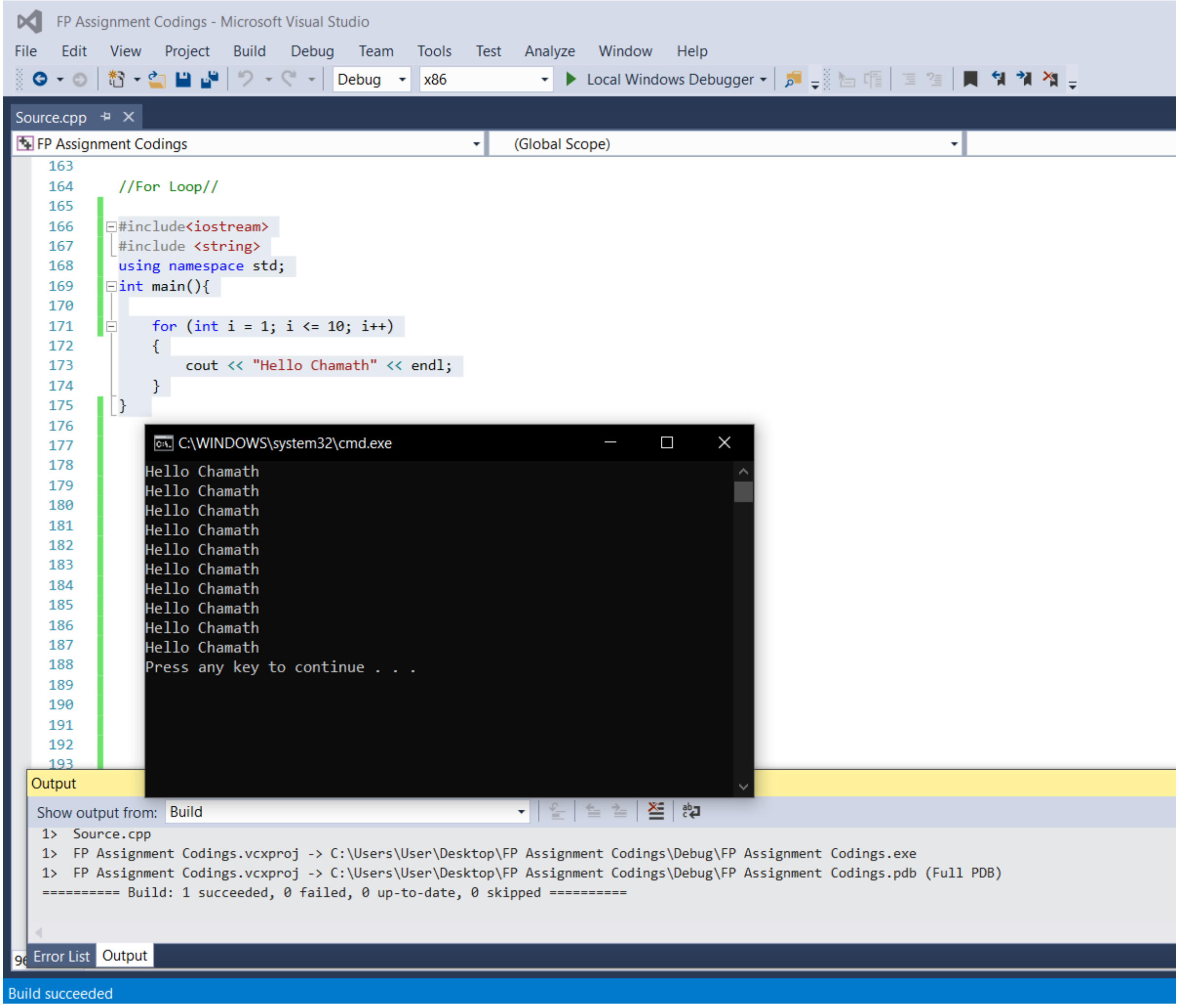
- `(number % 2 == 0)` – the condition
- `?` – represents if block in the if…else structure
- `cout<< "The number entered is even."` – statement
- `:` - represents else block in the if…else structure
- `cout <<"The number entered is odd.";` - statement

[1.2.2] Repetitions/Loops in programming

When it comes to loops, each and every modern high-level programming language bears a structure as loop. Through loops, it repeats of one or more instructions till a certain condition meets. Some occasions for using loops are to add sums of numbers, repeat functions, etc. For loop, while loop, do…while loop are few examples for loops.

1.2.2.1- 'For' Loop

This repetition structure is a loop which can be used when the number of repetitions is well known before entering the loop. This includes three steps as initialization, condition checking and updating. So, in here, initialization means variables and executed only one time at the beginning. Then the expression is checked and if its true, the body of the loop will be executed. But if the checked expression is false, for loop will be terminated and update expression will be updated. Thus, this process will be repeated till testing expression becomes false.


FP Assignment Codings - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp
FP Assignment Codings
(Global Scope)
C:\WINDOWS\system32\cmd.exe
Output
Show output from: Build
Error List Output
Build succeeded

```
//For Loop//

#include<iostream>
#include <string>
using namespace std;
int main(){

    for (int i = 1; i <= 10; i++)
    {
        cout << "Hello Chamath" << endl;
    }
}
```

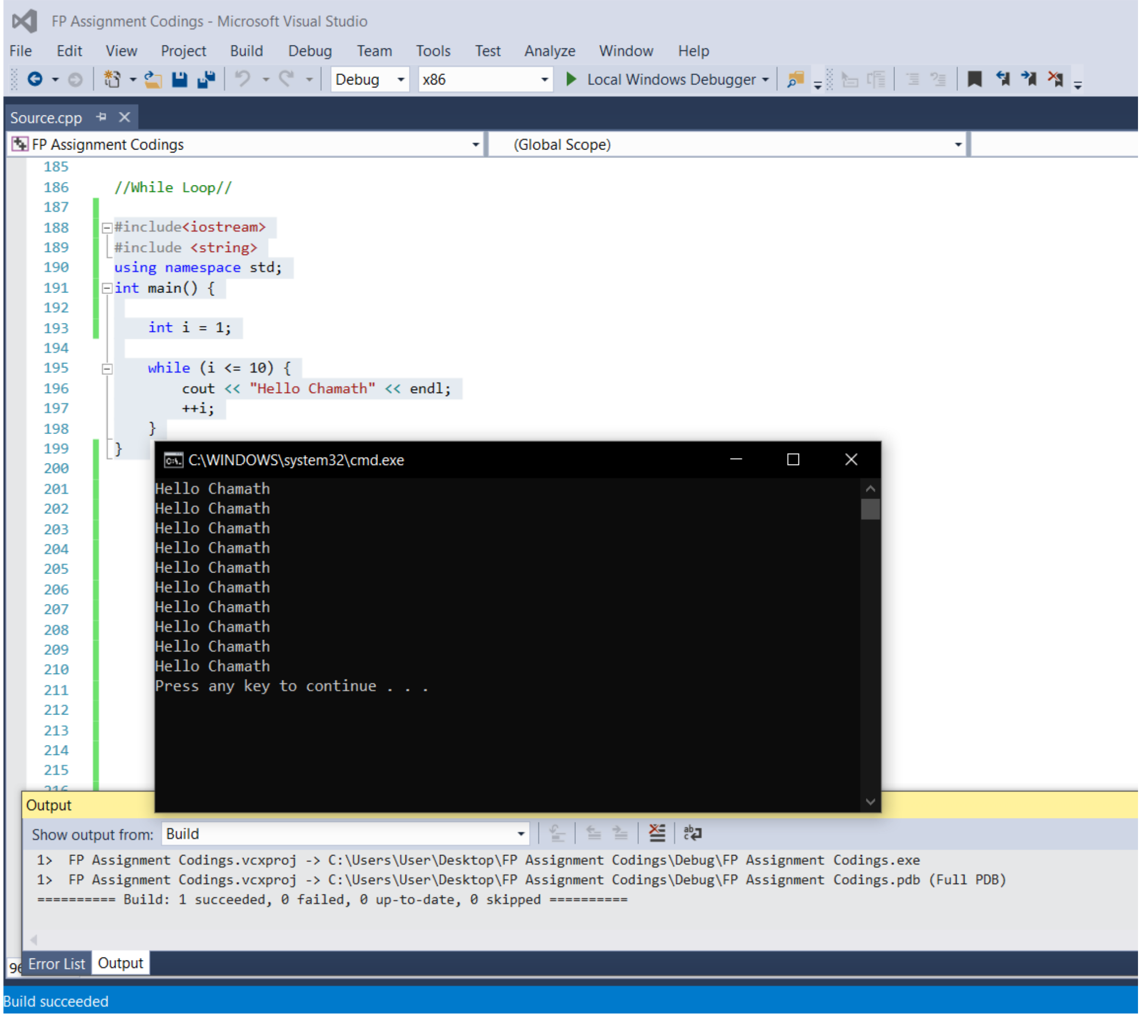
```
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Press any key to continue . . .
```

```
1>  Source.cpp
1>  FP Assignment Codings.vcxproj -> C:\Users\User\Desktop\FP Assignment Codings\Debug\FP Assignment Codings.exe
1>  FP Assignment Codings.vcxproj -> C:\Users\User\Desktop\FP Assignment Codings\Debug\FP Assignment Codings.pdb (Full PDB)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

As shown in the above screenshot of coding, `Hello Chamath` for only 10 times. In there, `i` means 'counter'. Thus, the above screenshot clearly depicts how the for loop works.

1.2.2.2- ‘While’ Loop

Simply, the while loop is repeating a specific statement(s) while a given condition becomes true. It doesn’t matter how many times it should be repeated, it will be repeated till the given condition is met. When only the given condition becomes false, the statement(s) will stop executing/ while loop will be terminated.


FP Assignment Codings - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp
FP Assignment Codings
(Global Scope)
C:\WINDOWS\system32\cmd.exe
Output
Show output from: Build
Error List
Output
Build succeeded

```
//While Loop//

#include<iostream>
#include <string>
using namespace std;
int main() {

    int i = 1;

    while (i <= 10) {
        cout << "Hello Chamath" << endl;
        ++i;
    }
}
```

```
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Hello Chamath
Press any key to continue . . .
```

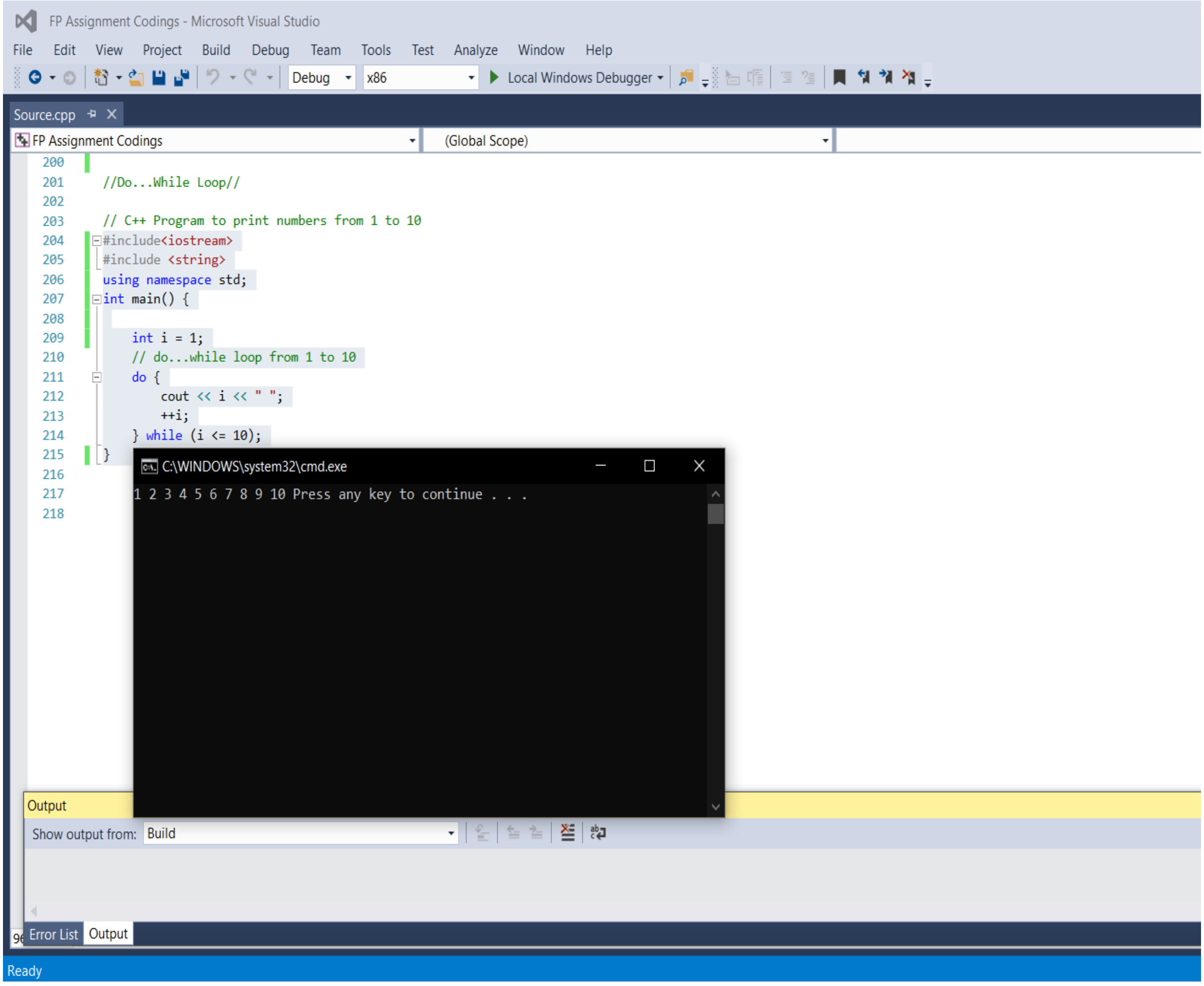
```
1>  FP Assignment Codings.vcxproj -> C:\Users\User\Desktop\FP Assignment Codings\Debug\FP Assignment Codings.exe
1>  FP Assignment Codings.vcxproj -> C:\Users\User\Desktop\FP Assignment Codings\Debug\FP Assignment Codings.pdb (Full PDB)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

As shown in the above screenshot of coding, `Hello Chamath` for 10 times while counter or `i` equals or less than 10. Thus, the above screenshot clearly depicts how the while loop works.

1.2.2.3- ‘Do…While’ Loop

This loop is also called as an exit-controlled loop The Do…While Loop just like the while loop. Thus, the Do…While Loop is a variant of the while loop with one important difference.

That difference is that always the condition is checked after the body of the loop which means it checks the body of the loop at first and condition at the bottom. If the given condition is true, the control flow is reactivated and the loop instruction(s) is executed again. This process is repeated till the given condition is false. When the given condition is false, the looping will be stopped.



FP Assignment Codings - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp
FP Assignment Codings
(Global Scope)
C:\WINDOWS\system32\cmd.exe
Output
Show output from: Build
Error List
Output
Ready

```
//Do...While Loop//

// C++ Program to print numbers from 1 to 10
#include<iostream>
#include <string>
using namespace std;
int main() {

    int i = 1;
    // do...while loop from 1 to 10
    do {
        cout << i << " ";
        ++i;
    } while (i <= 10);
}
```

```
1 2 3 4 5 6 7 8 9 10 Press any key to continue . . .
```

As shown in the above screenshot of coding, at first the body of the program is checked which means the statements inside the do is checked at fist and after that only the condition which means the statement under while condition will be checked. Thus, the above screenshot clearly depicts how the do…while loop works.

1.3) Modularization in programming

Primarily, it's a must to understand what is a module in programming before explaining about modularization. Modules means the basic unit in software development and operation and can use modules to create a solution that can be implemented with a single instruction.

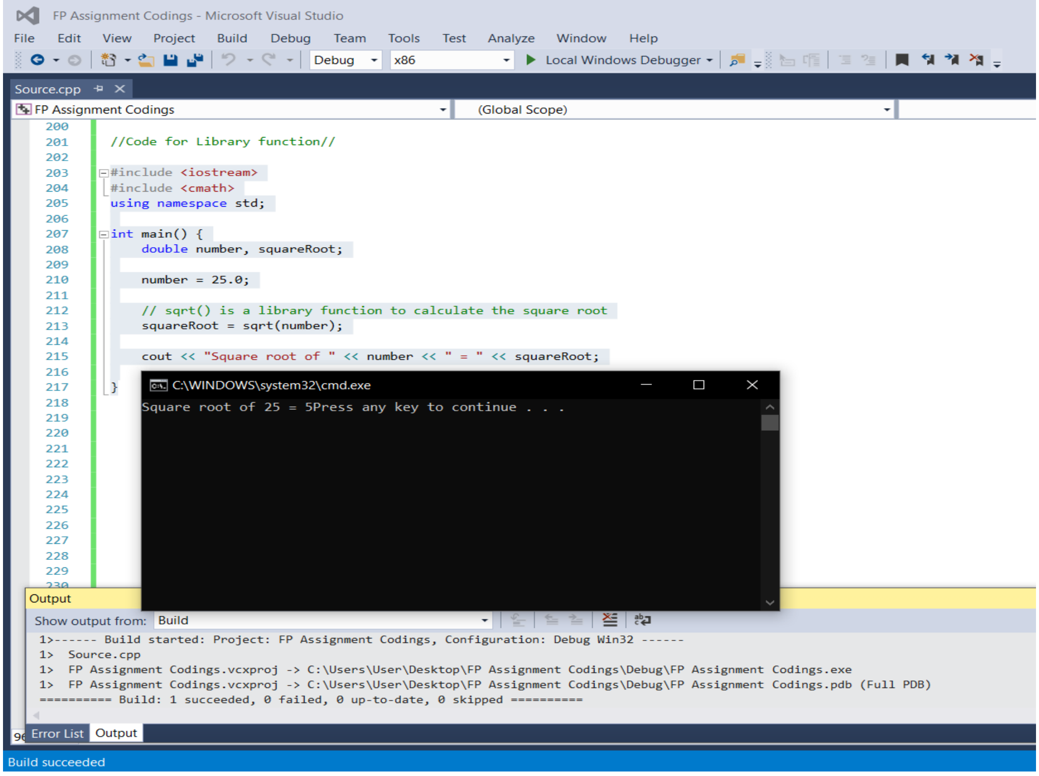
And also, can reuse the modules which are created. As the functional requirements increase consistently, the software system becomes complex and larger. Therefore, one of the key functions of the software design process is to divide the software specification into several program modules that simultaneously meet the key problem description. Thus, its crucial for developers/programmers to determine the organization of modules which means the grouping of modules that attain a particular function. Key conceptual basis for modularization includes strengths, data masks/hiding, etc. So, it is common that a good modularized software system can be easily understood, developed and managed. Further, modularization concept is varied due to the programming methodology used by the programmer. For instance, in SOP (Structured Oriented Programming) modules are called as functions while in OOP (Object Oriented Programming) modules are called as methods. However, the main goal of modularization is to achieve a complete structure of the software system according to a set of predefined rules.

An example where modularization concept used is a simple calculator with adding, subtracting, multiplying and dividing functions.

1.3.1- Functions

1.3.1.1 - Library Functions

These Library functions also called as Built-in functions and Pre-defined functions. Library functions are grouped predefined functions and available as libraries along with the programming language itself. So, each and every library function perform certain operations. Library functions are varied from one programming language to the other. Instead of writing programmer's own code to get particular outputs, can use the library functions to get the pre-defined output. These library functions contain within header files provided with the compiler. Few examples for library functions in C++ language are 'concat', 'strlen' and 'sqrt'.



FP Assignment Codings - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp
FP Assignment Codings
(Global Scope)

```
//Code for Library function//

#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double number, squareRoot;

    number = 25.0;

    // sqrt() is a library function to calculate the square root
    squareRoot = sqrt(number);

    cout << "Square root of " << number << " = " << squareRoot;

}
```

C:\WINDOWS\system32\cmd.exe

```
Square root of 25 = 5Press any key to continue . . .
```

Output

Show output from: Build

```
1>------ Build started: Project: FP Assignment Codings, Configuration: Debug Win32 ------
1>  Source.cpp
1>  FP Assignment Codings.vcxproj -> C:\Users\User\Desktop\FP Assignment Codings\Debug\FP Assignment Codings.exe
1>  FP Assignment Codings.vcxproj -> C:\Users\User\Desktop\FP Assignment Codings\Debug\FP Assignment Codings.pdb (Full PDB)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Build succeeded

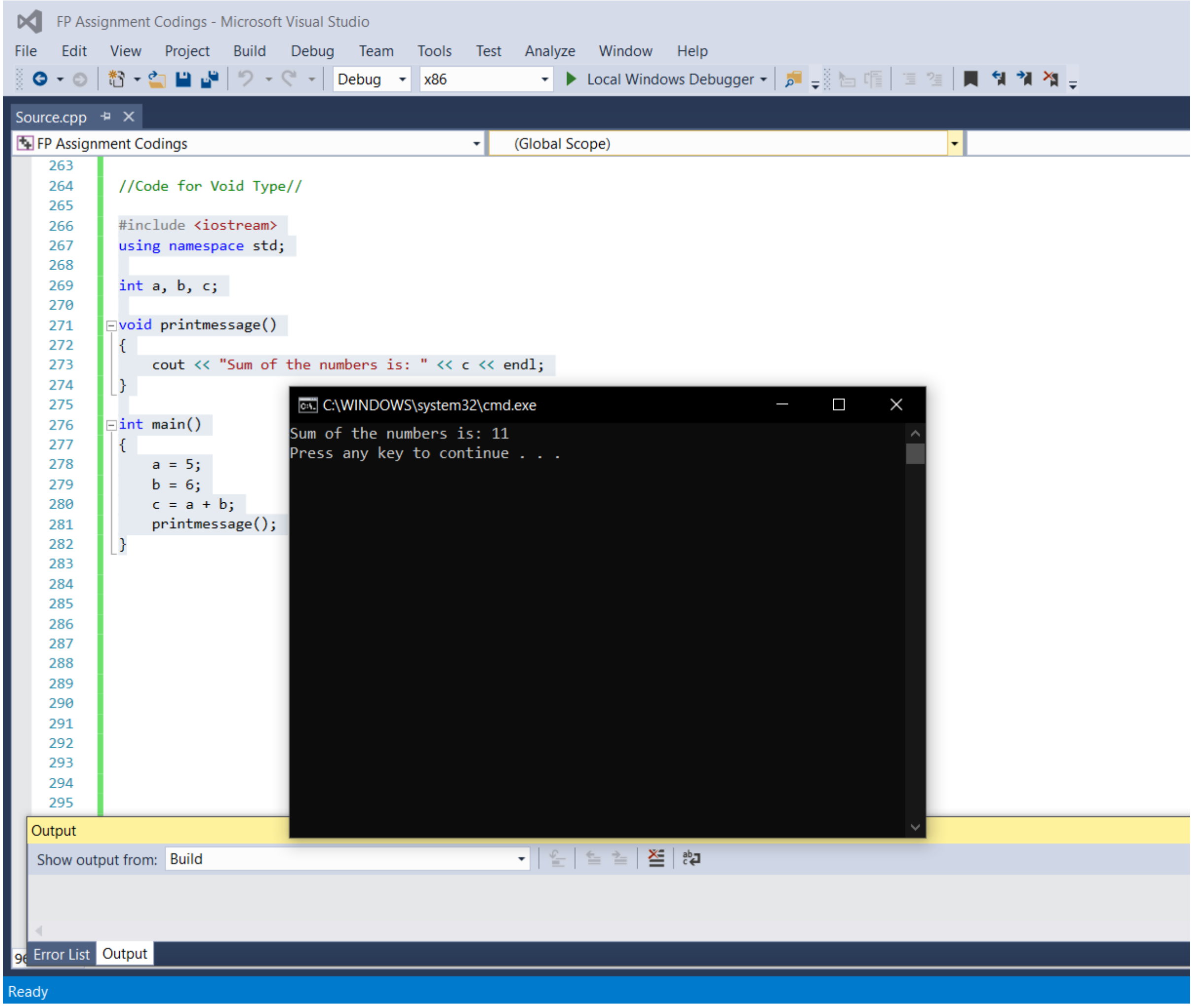
1.3.1.2 - User Defined Functions

When it comes to user-defined functions, these types of functions are used by programmers when library functions in the programming language cannot satisfy the programmers intended requirements. Therefore, its clear that user-defined functions are used by programmers for specific tasks. So, the programmer has to build up a particular own function in the program to satisfy the requirements which cannot be satisfied using library functions. These kinds of functions can be repeatedly use in the program wherever it is necessary to use. And also, it helps in reducing the complexity of the program.

Further, User-Defined functions can be written according to two methods basically. Below explanations will be focused on those two methods.

- Void Functions (User-defined functions with no return values)

Simply, it’s convenient to say that void functions are used when there’s no return value after the function execution. These void functions use the ‘void’ keyword. Further, there are some void functions which display information for user to read and some void functions are used with ‘reference’ parameter like arrays.



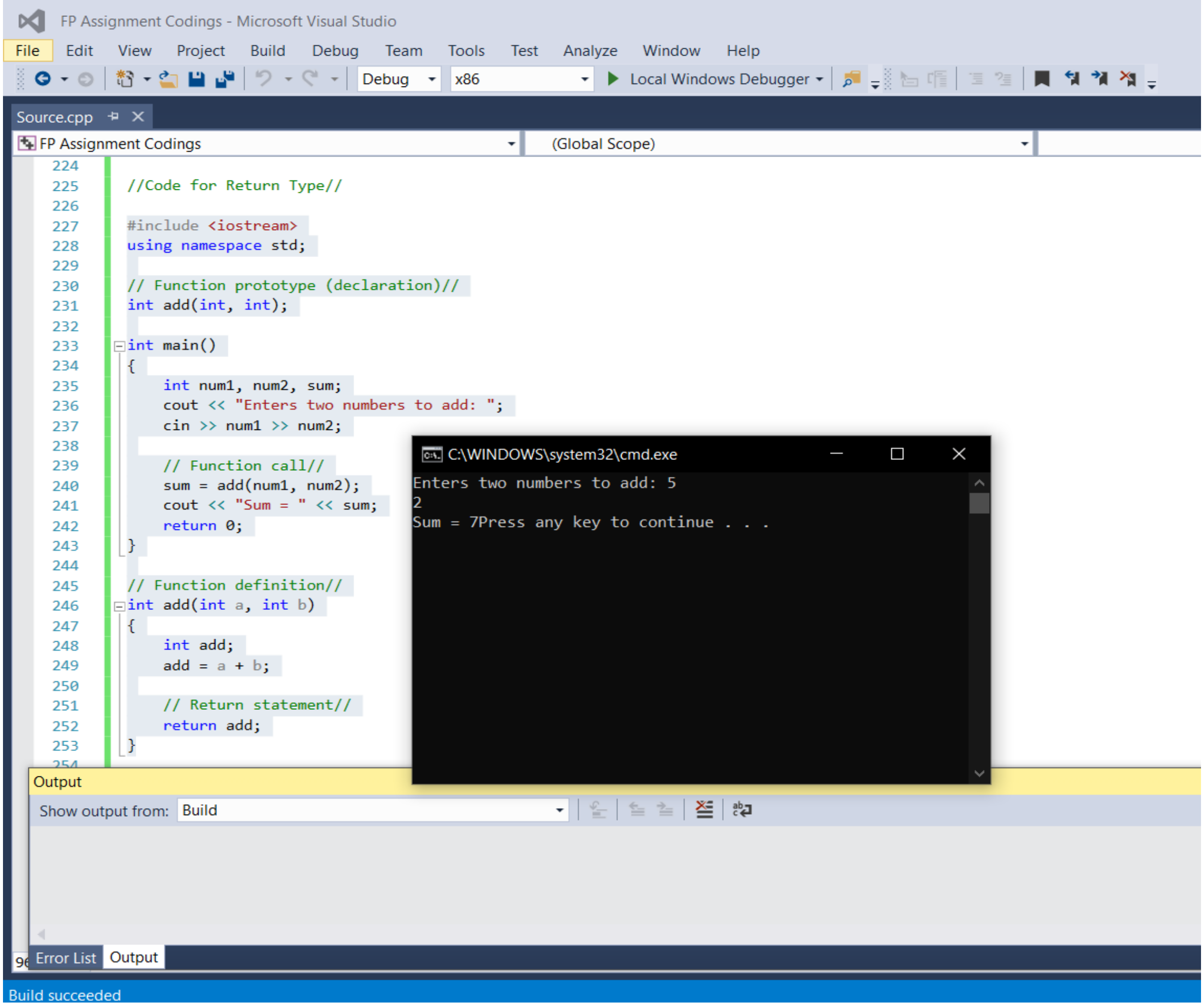
FP Assignment Codings - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp
FP Assignment Codings
(Global Scope)
C:\WINDOWS\system32\cmd.exe
Output
Show output from: Build
Error List
Output
Ready

```
//Code for Void Type//

#include <iostream>
using namespace std;

int a, b, c;

void printmessage()
{
    cout << "Sum of the numbers is: " << c << endl;
}

int main()
{
    a = 5;
    b = 6;
    c = a + b;
    printmessage();
}
```

```
Sum of the numbers is: 11
Press any key to continue . . .
```

- Return Type Functions (User-defined functions with return values)

In situations where it should return values to another different function like to the main function, return type functioned are used. Thus, return type functions can be used in output, assignment and arguments in other function calls while void functions can be used only for stand-alone statements.



```
//Code for Return Type//

#include <iostream>
using namespace std;

// Function prototype (declaration)//
int add(int, int);

int main()
{
    int num1, num2, sum;
    cout << "Enters two numbers to add: ";
    cin >> num1 >> num2;

    // Function call//
    sum = add(num1, num2);
    cout << "Sum = " << sum;
    return 0;
}

// Function definition//
int add(int a, int b)
{
    int add;
    add = a + b;

    // Return statement//
    return add;
}
```

```
Enters two numbers to add: 5
2
Sum = 7Press any key to continue . . .
```

Task 2 (System Design)

2.1) SRS Documentation

Software Requirement Specification (SRS) for Billing System in Rathnayaka GYMS

Prepared by:
CL/HDCSE/95/
43
Date Created:
06/04/2021

1. Introduction

1.1 Purpose

Making an application which can automate Rathnayaka GYMS fitness center's billing process is to make it easier their billing process. Viewing of details and buying of supplements and equipment from the gym stores and selecting of fitness packages by the customers and more importantly the calculation of bills is considered here. It provides environment to maintain the customer details starting from getting new connection, receiving bill, payments etc. People who are going to use the system are:

- User type 1-administrator
- User type 2-cashier

Billing is done by collection of payments of supplements, packages, equipment and there by generating receipts for each consumer, receipt cancel in the case of errors. Proposed system will provide authentication to avoid unauthorized access.

The purpose of this Software Requirements Specification document is to maintain all the functions and the specifications of 'Rathnayaka GYMS's Billing System'. Besides it contains detailed descriptions of all the requirements specified. It will explain the purpose and features of the billing system, the interfaces of the system, what the system will do, and the constraints under which it must operate.

1.2 Document Conventions

In this document I used Times New Roman font style and font size as 12, headings are bold and bullet points are used in order to prepare this document more effective.

1.3 Intended Audience and Reading Suggestions

This intended audience of this document includes staff of the Rathnayaka GYMS. In staff also, this will be intended to administrator and cashier. The rest of this SRS contains overall description, system feature, external interface requirements, other non-functional requirements and other requirements. Programmers, testers, etc. are few example readers of this system requirement specification document. Those readers should be able to refer this system requirement specification document and design, implement, test the system according to requirements. And also, to test the functions of the system.

1.4 Project Scope

The purpose of this application is to develop and automate the Billing System at Rathnayaka GYMS, which is an application that provides a service to the administrator and cashier to deal with transactions and make it easier their works by reducing maintaining bill books and books which include package details, food supplement details and equipment details. This system becomes a service for customers as it less consumes customers' valuable time.

Features are in the scope of the software are:

- View Fitness Package Details
- View Food Supplement Details
- View Fitness Equipment Details
- Calculate Bill
- About Us.
- Current prices of supplements, packages, equipments
- Authentication level (admin, cashier)

1.5 References

https://uccs.edu/Documents/tboult/srs.doc

2. Overall Description

2.1 Product Perspective

The system will allow to monitor software specification documents related to sales, stocks, packages at any time which are currently done in manually and this will minimize the causing of error too. Thus, this project of Billing System is used by two users and they are administrator and cashier. The overall control of this billing system is having only with administrator. Cashier also can log in to the system and perform certain functions which are relevant. Simply, Administrator have the fullest authorization to view all payment details, package details etc,. In this system, the admin should have separate ID, password and cashier should have separate ID, password. Billing System is intended to be a stand-alone application. It should run only on Windows based platform. The Billing System for Rathynayaka GYMS will be user-friendly, and effective software application which will satisfy all the above purposes.

2.2 Product Functions

2.2.1 Correctness of the Login Credentials

Initializes the system for Admin/Cashier Authentication step by getting the unique login credential assigned for users.

2.2.2 System Authorization/Login

Log in to the Billing System by providing the ID and Passwords.

2.2.3 View fitness package details, supplement details and equipment details

The system allows user to view fitness packages, view food supplements and view fitness equipment which are available in the gym.

NOTE: Prices of the Membership Rate and the Introductory Rate are same!

2.2.3 Calculation of Bill

Finalizing the bill by ending fitness membership fees, supplement charges, equipment charges transactions and deducting discount of 0.5 % which exceeds the total amount of Rs. 5000

2.2.4 Issuing of receipt

When issuing the bill, it would display invoice number, customer’s name, date, names of the services which goes with transactions done, prices, deductions and Rathnayaka gym’s slogan (RATHNAYAKA GYMS-Beyond Fitness). Further, cancellations of bills (exiting the system) allow user to recorrect the mistakes done in the receipt and to re prepare it from the top.

2.3 User Characteristics

The users (admin and cashier) should be familiar with the Billing process. Users should be good at adding, subtracting, dividing, etc. and should know basic stuffs in computing and calculations. Specially users should process the knowledge of how to do a transaction, generate bills and create receipts.

2.4 Operating Environment

Windows OS will be used in operating this software.

2.5 Design and Implementation Constraints

- Hardware Limitations:

Processor	Intel core i3 or i5
RAM	16GB
VGA	2 GB or 8GB
Hard Disk	1TB
Optical Drive	SSD or DVD ROM
Operating System	Windows 10
Software	VS (Visual Studio)
Basic Requirements	Laptop, Any Monitor, Mouse, Keyboard, Printer like basic stuffs

- Interfaces to other application:

Different interfaces for viewing of details is provided and also a different interface (various functions) for bill payment is provided.

- Security Considerations:

The password and ID are the security functions are there to be considered here. The data protection will be done by the backups.

- Criticality of the Application: The servers should be available during 365 days.

3. Specific Requirements

3.1 Functional Requirements

Simply, in here functional requirements include what the billing system does and means about the software system features.

3.1.1 Initializations/ Providing Login Credentials

3.1.1.1 Admin or Cashier initiates Billing System by providing relevant login credential.
3.1.1.2 In the home directory window, all the working files will be generated. Admin or Cashier is also asked for the initial login and password.
3.1.1.3 Admin or Cashier specifies the home directory window and login/password.

3.1.2 Authorization/Login

3.1.2.1 Administrator's Login- By giving the administrator's ID and password the administrator can login into the Billing System and thereby manage the bill and view fitness package, food supplement and equipment details.
3.1.2.2 Cashier's Login- By giving the cashier's ID and password the cashier can login into the Billing System and thereby engage in viewing details & issuing bills.
3.1.2.3 If Cashier's or Admin's Authorization failed- Remind that the user has typed the wrong password and let the user exit the system.

3.1.4 Calculation of Bill

3.1.4.1 Check for the food supplements which are bought, selected fitness packages, check for the fitness equipment which are bought and add the ID/Code of those selected items/packages to the bill along with their prices.
3.1.4.2 Check the total items (number of food supplements and number of fitness equipment) bought and multiply it by the quantity
3.1.4.3 Calculate the total charges of the food supplement items bought (if selected to buy), fitness equipment bought (if selected to buy) and also add charges of fitness packages if selected.
3.1.4.4 Calculate the total bill and deduct a discount of 0.5% only if total amount exceeds Rs.5000.
3.1.4.5 Finally calculate the final total amount to be paid by the customer.

3.1.5 Issuing of receipt

3.1.5	**Property**	**Should display or not**	**Explanation**
3.1.5.1	Rathnayaka GYMS Slogan	Yes	The name ‘RATHNAYAKA GYMS’ and under that name their slogan ‘*Beyond Fitness*’ should be displayed in the right-side top.
3.1.5.2	Invoice Number	Yes	Invoice number should be displayed at the left-side top.
3.1.5.3	Customer Name	Yes	Under the invoice number, customer name should be displayed
3.1.5.4	Date	Yes	Under the customer’s name, date should be displayed.
3.1.5.5	Time	Yes	Under the date, time should be displayed.
3.1.5.6	Food Supplement	Yes	In the middle of the receipt, there should be a select option to select the food supplement which display with the price after selecting.
3.1.5.7	Fitness Package	Yes	Under the selected supplement, there should be an option to select the package which display with the price after selecting.
3.1.2.8	Fitness Equipment	Yes	Under the selected package, there should be an option to select the equipment which display with the price after selecting.
3.1.5.9	Quantity	Yes	After selecting the food supplement or the fitness equipment, quantity of the bought item(s) should be entered.
3.1.5.10	Total Amount	Yes	All the total values of the packages, food items and equipment bought (after multiplying with the quantity) should be displayed next.
3.1.5.11	Discount Earned	Yes	Under the total amount, Discount amount earned should be displayed if the total exceeds Rs.5000.
3.1.5.12	Final Total Due Amount to be paid	Yes	Under the discount earned, total due amount to be paid by the customer after deducting the discount earned should be displayed.
3.1.5.13	Paid(cash) Amount	Yes	Under total due amount to be paid, the customer paid amount should be displayed.
3.1.5.14	Balance Given	Yes	Under the paid amount, the balance given to the customer should be displayed.
3.1.5.15	Payment Type	Yes	Under the balance given, payment type should be selected whether it’s a cash payment or card payment.

3.2 Non-Functional Requirement

Simply in here, Non-functional requirements mean requirements that can be used to describe operation of the Billing System. Not the specific behaviors or what system does. Execution qualities, like security and usability, are observable at run time of the program.

3.2.1 Performance

It’s expected the system to be accessible 365 days (every time) and because of that the server also should be available all the time. This Billing System will be implemented with C++ using the Visual Studio Integrated Development Environment and as the back end. Procedures like login, receipt issuing should be done quickly within 3 to 5 seconds while generating the outputs of the calculations also with a minimum time duration. This System should run-on 32-bit OS or 64-bit OS.

3.2.2 Availability

This Billing System should be available within all the hours and days (every time).

3.2.3 Reliability

Here, the confidentiality of the data is highly considered. The access for log into this system must have only for administrator and cashier. Only the admin should have the access to each and every function in this system while cashier have limited functions. Not needed to wait for a long time to login.

3.2.4 Security

The access to the billing application will be restricted to the authorized users who are identified by a valid ID and password. The users can login to the system using the ID and password and access the functions based on the role which they are assigned.

3.2.5 Safety Requirements

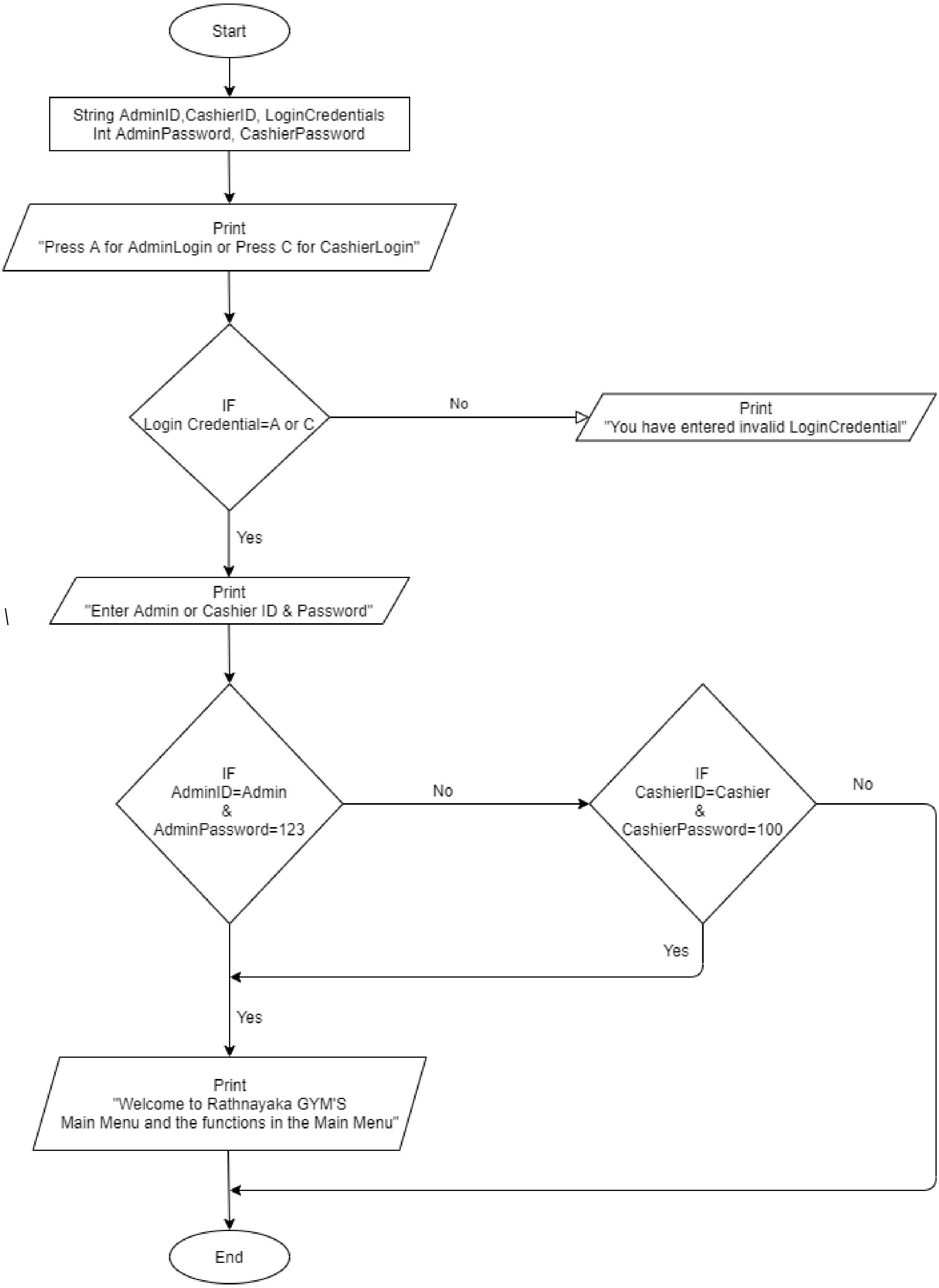
An error message should be displayed when a user entered incorrect login credential, username/password and should provide the user to exit.

3.2.6 Maintenance

Maintaining, simply which means using of the system is easier and cost for maintenance is low.

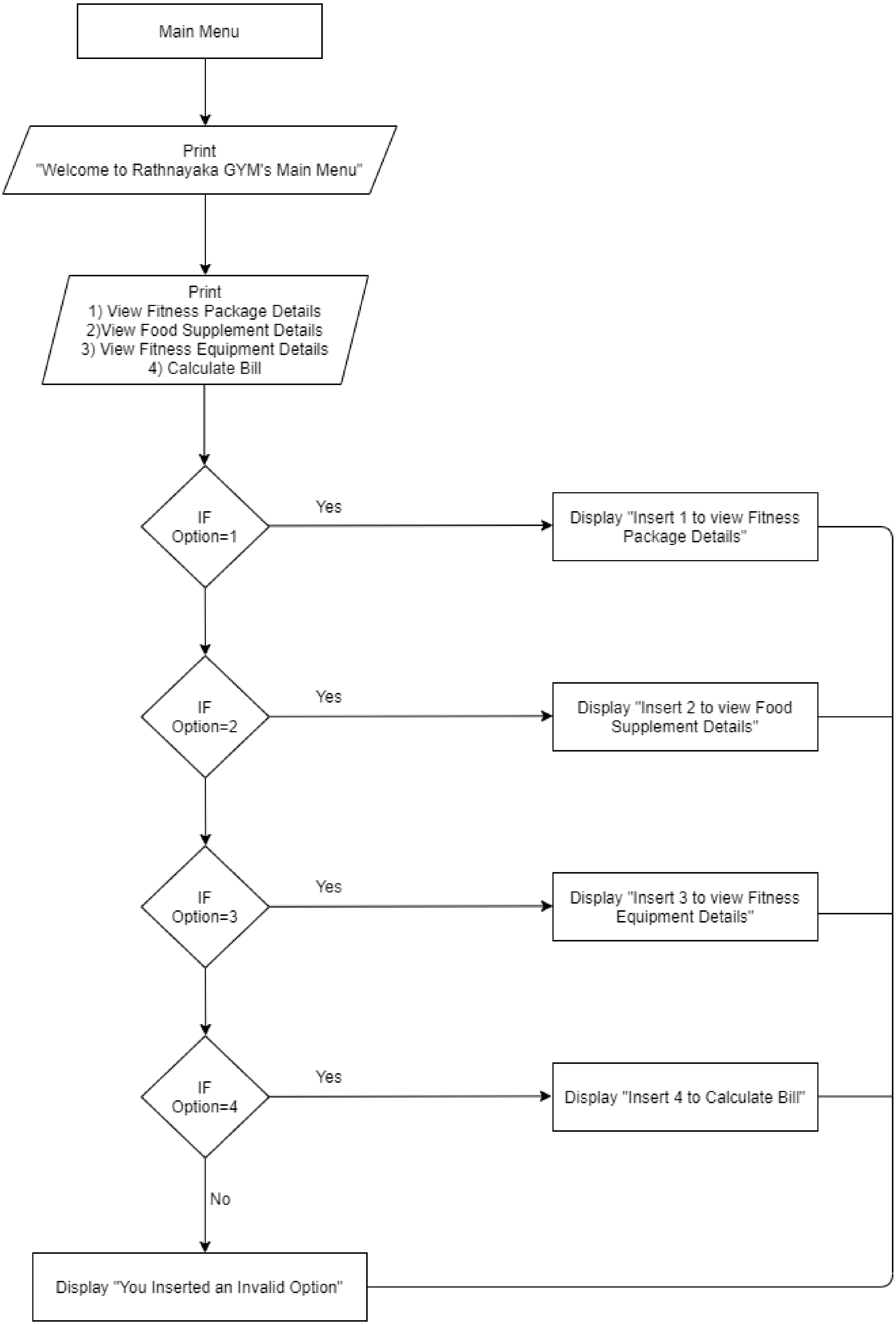
2.2) Flow charts

- For Login

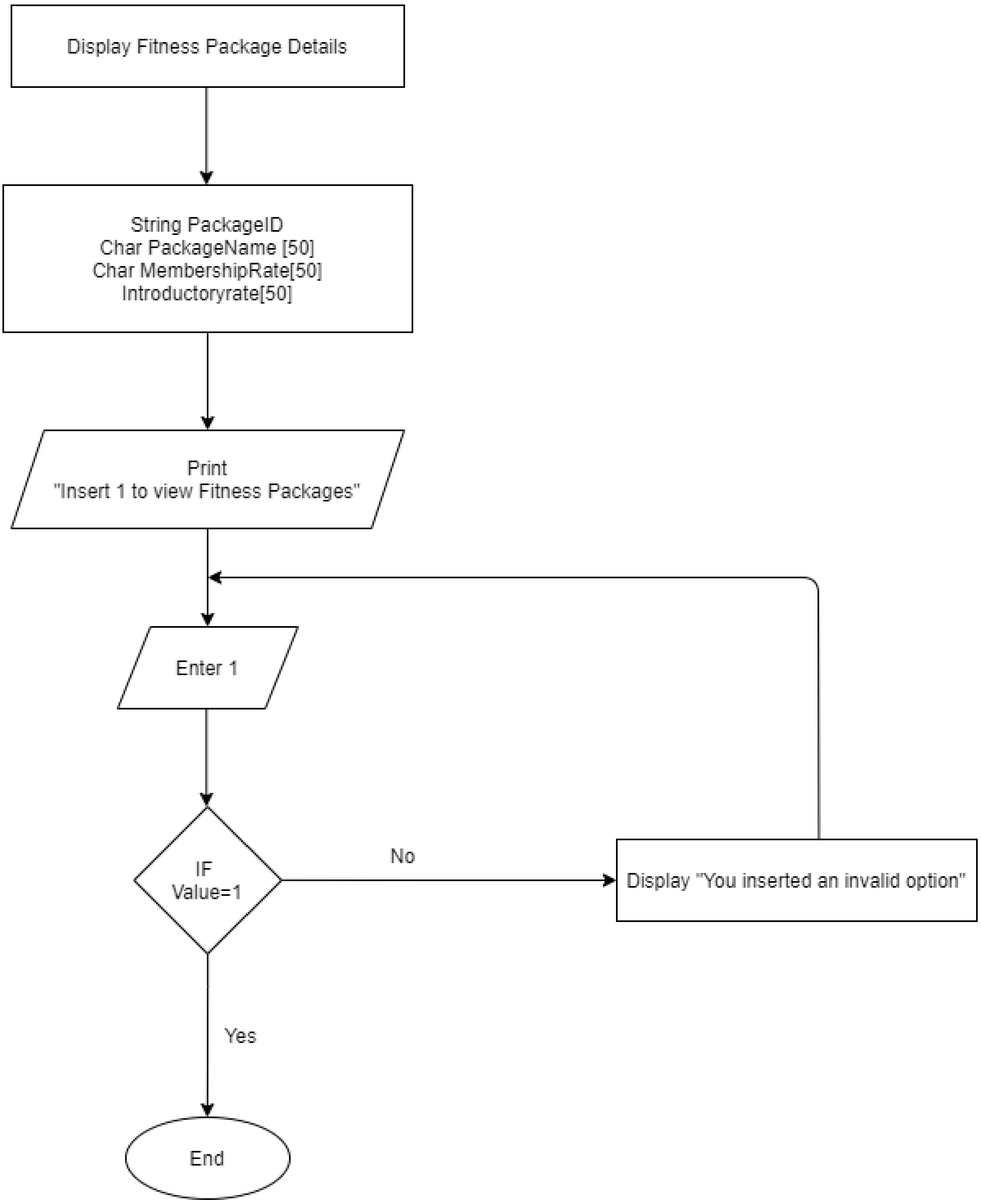

Start
String AdminID,CashierID, LoginCredentials
Int AdminPassword, CashierPassword
Print
"Press A for AdminLogin or Press C for CashierLogin"
IF
Login Credential=A or C
No
Print
"You have entered invalid LoginCredential"
Yes
Print
"Enter Admin or Cashier ID & Password"
IF
AdminID=Admin
&
AdminPassword=123
No
IF
CashierID=Cashier
&
CashierPassword=100
No
Yes
Yes
Print
"Welcome to Rathnayaka GYM'S
Main Menu and the functions in the Main Menu"
End

\

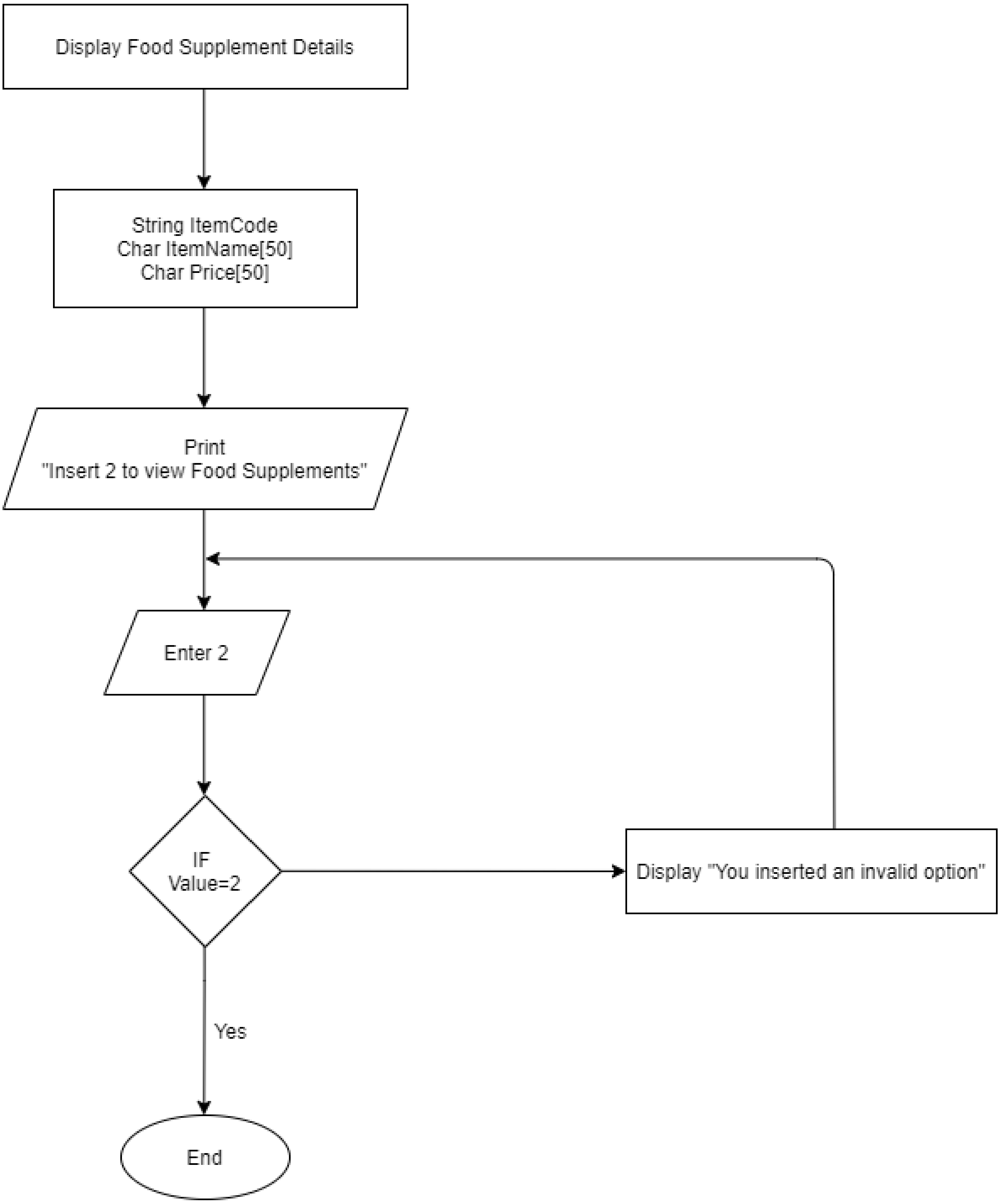
- For Main Menu


Main Menu
Print
"Welcome to Rathnayaka GYM's Main Menu"
Print
1) View Fitness Package Details
2)View Food Supplement Details
3) View Fitness Equipment Details
4) Calculate Bill
IF
Option=1
Yes
Display "Insert 1 to view Fitness Package Details"
IF
Option=2
Yes
Display "Insert 2 to view Food Supplement Details"
IF
Option=3
Yes
Display "Insert 3 to view Fitness Equipment Details"
IF
Option=4
Yes
Display "Insert 4 to Calculate Bill"
No
Display "You Inserted an Invalid Option"

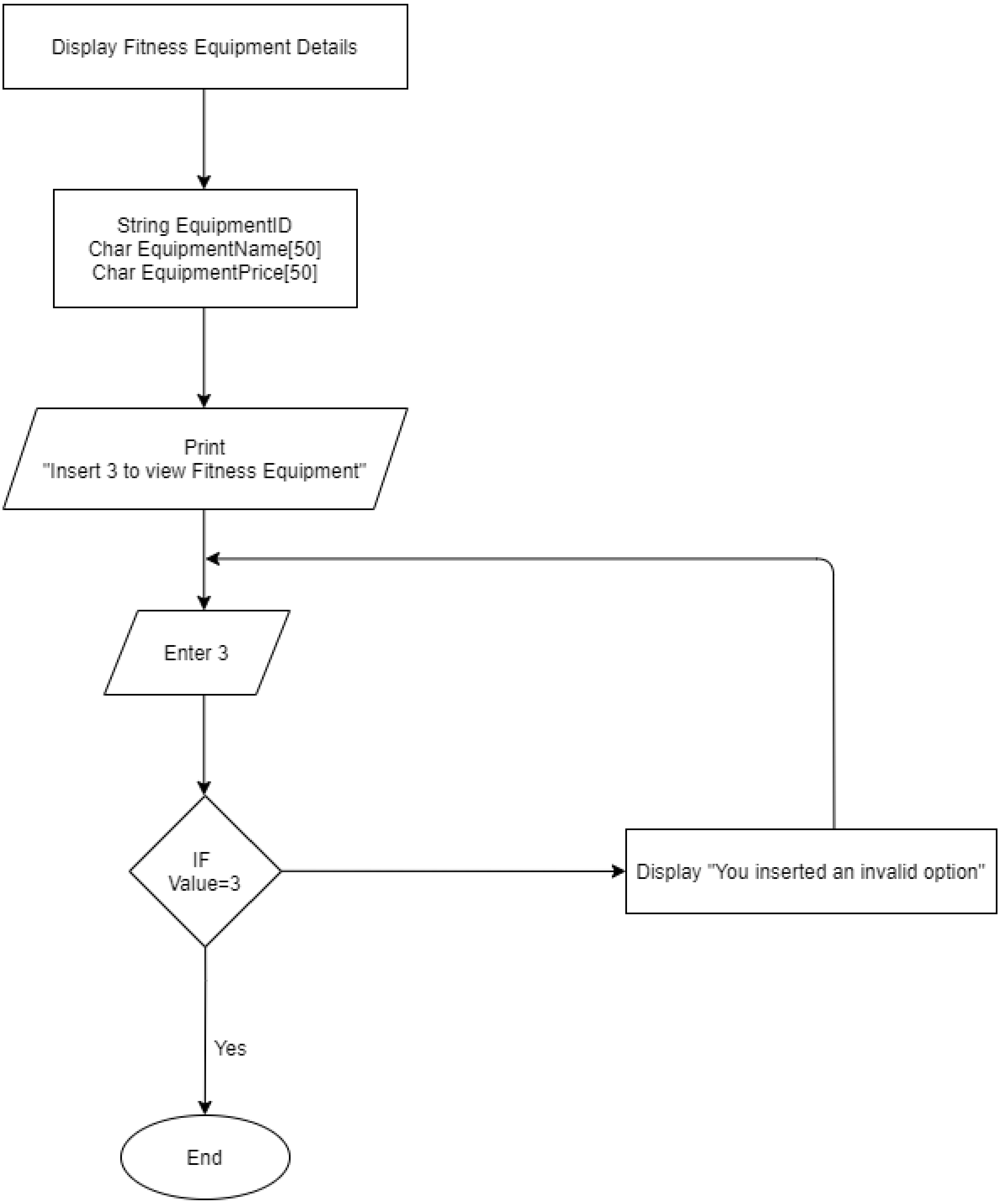
- To View Fitness Packages



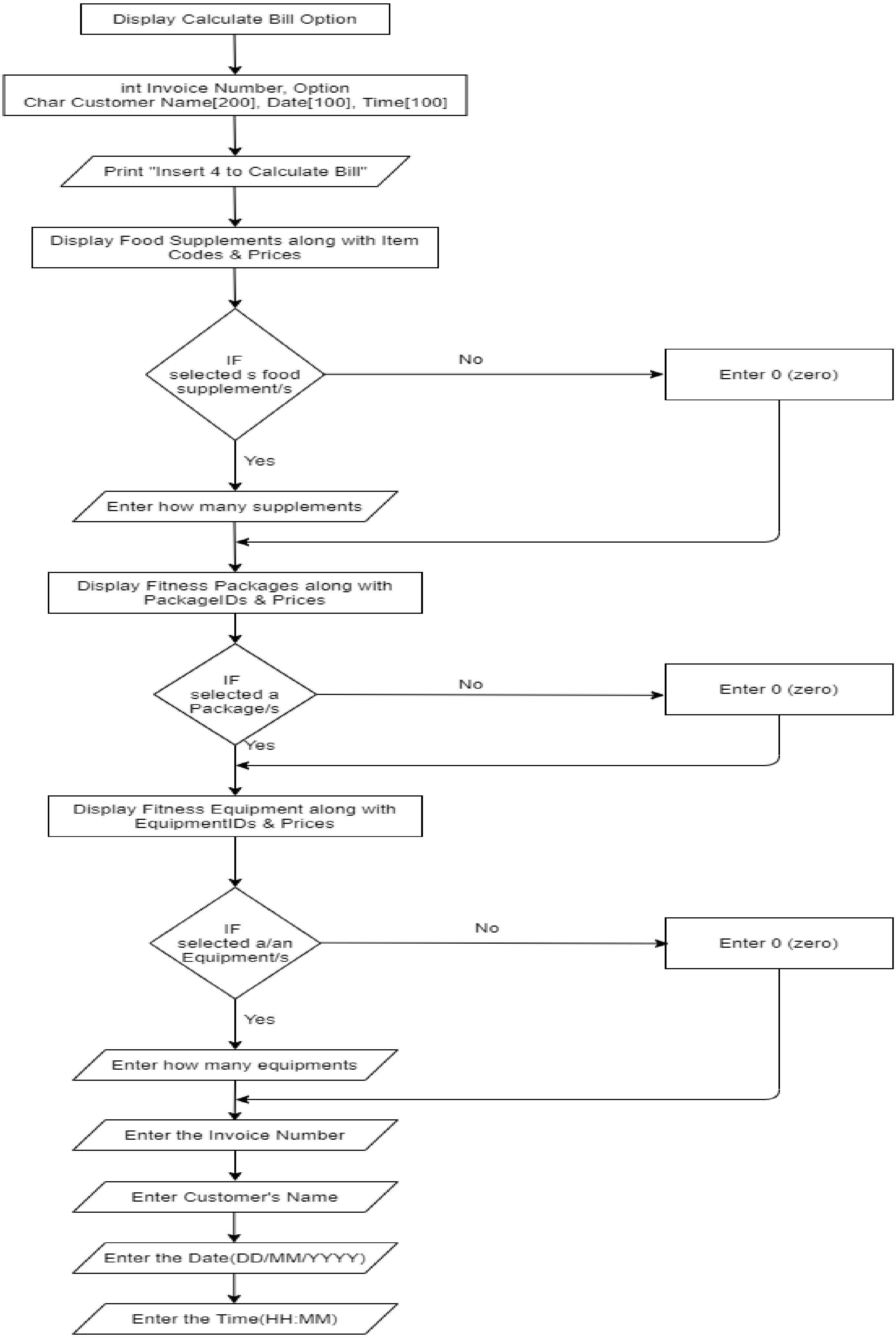
- To View Food Supplements


Display Food Supplement Details
String ItemCode
Char ItemName[50]
Char Price[50]
Print
"Insert 2 to view Food Supplements"
Enter 2
IF
Value=2
Display "You inserted an invalid option"
Yes
End

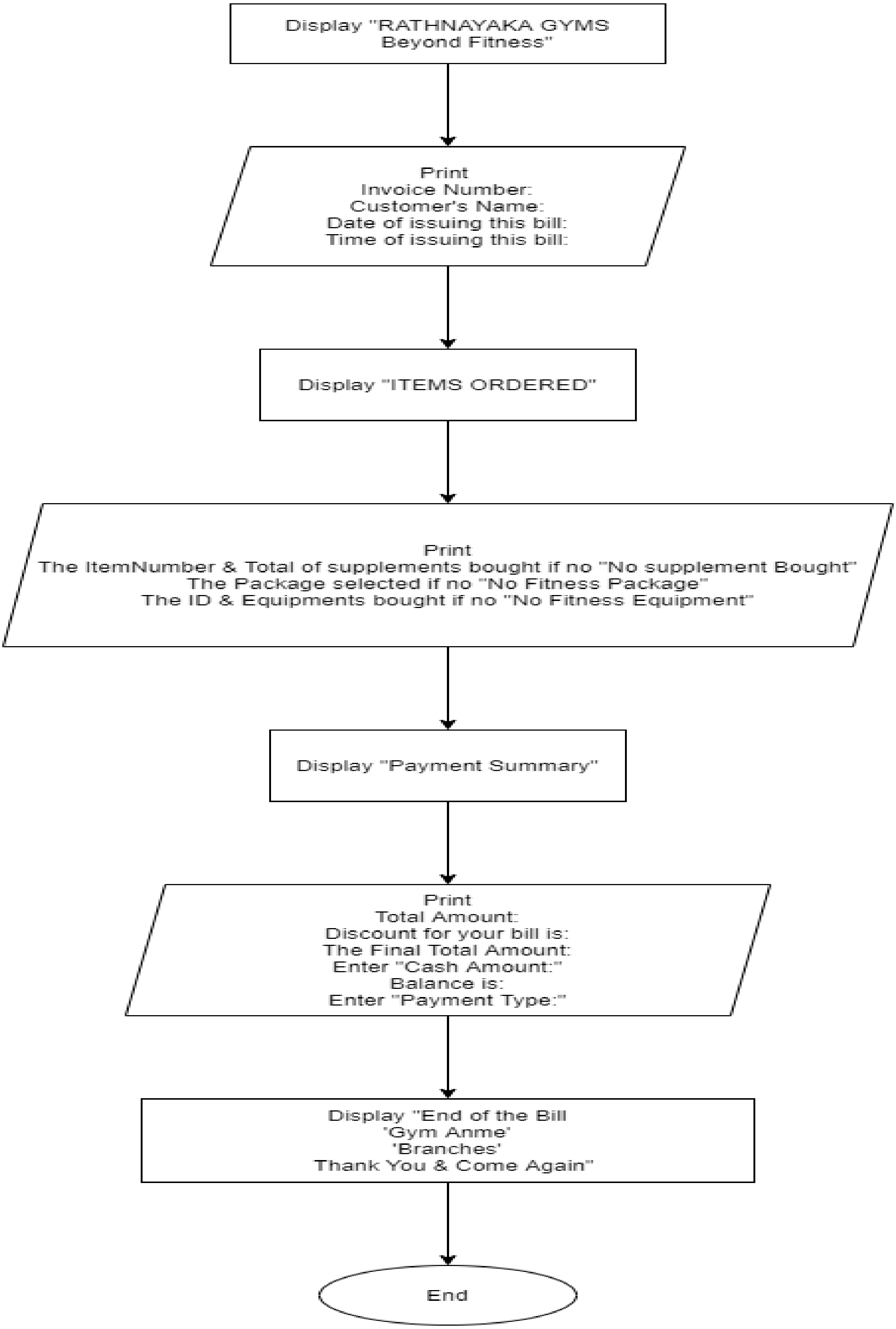
- To View Fitness Equipment


Display Fitness Equipment Details
String EquipmentID
Char EquipmentName[50]
Char EquipmentPrice[50]
Print
"Insert 3 to view Fitness Equipment"
Enter 3
IF
Value=3
Display "You inserted an invalid option"
Yes
End

- To Calculate Bill


Display Calculate Bill Option
int Invoice Number, Option
Char Customer Name[200], Date[100], Time[100]
Print "Insert 4 to Calculate Bill"
Display Food Supplements along with Item Codes & Prices
IF selected s food supplement/s
No
Enter 0 (zero)
Yes
Enter how many supplements
Display Fitness Packages along with PackageIDs & Prices
IF selected a Package/s
No
Enter 0 (zero)
Yes
Display Fitness Equipment along with EquipmentIDs & Prices
IF selected a/an Equipment/s
No
Enter 0 (zero)
Yes
Enter how many equipments
Enter the Invoice Number
Enter Customer's Name
Enter the Date(DD/MM/YYYY)
Enter the Time(HH:MM)

- Issuing of Bill


Display "RATHNAYAKA GYMS
Beyond Fitness"
Print
Invoice Number:
Customer's Name:
Date of issuing this bill:
Time of issuing this bill:
Display "ITEMS ORDERED"
Print
The ItemNumber & Total of supplements bought if no "No supplement Bought"
The Package selected if no "No Fitness Package"
The ID & Equipments bought if no "No Fitness Equipment"
Display "Payment Summary"
Print
Total Amount:
Discount for your bill is:
The Final Total Amount:
Enter "Cash Amount:"
Balance is:
Enter "Payment Type:"
Display "End of the Bill
'Gym Anme'
'Branches'
Thank You & Come Again"
End

2.3) Pseudo Codes

- For Login

```
BEGIN
CONSIDER VARIABLES AS LoginCredentials, AdminID, CashierID, AdminPassword, CashierPassword
DISPLAY ENTER LOGIN CREDENTIAS OPTION
INSERT LoginCredentials
IF(CONDITION)
TRUE
DISPLAY ENTER ID & PASSWORD OPTION
ELSE
    DISPLAY “You Have Entered Invalid LoginCredentials”
IF(CONDITION)
TRUE
   DISPLAY “AdminID:”
   DISPLAY “AdminPassword:”
INSERT AdminID & AdminPassword
DISPLAY MAIN MENU
ELSE IF
   DISPLAY “CashierID:” + CashierID
   DISPLAY “CashierPassword:” + CashierPassword
INSERT CashierID & CashierPassword
DISPLAY MAIN MENU
ELSE
    DISPLAY “You Have Entered Invalid ID or Password”
END
```

- For Main Menu

```
BEGIN
CONSIDER VARIABLES AS Option,1,2,3,4
INSERT Option
DISPLAY "1) To View Fitness Package Details   2) To View Food Supplement Details 3) To View Fitness Equipment 4) To Calculate Bill"
DECLARE Option 1
DECLARE Option 2
DECLARE Option 3
DECLARE Option 4
INSERT 1
IF (CONDITION)
TRUE
    DISPLAY ALL AVAILABLE FITNESS PACKAGE DETAILS
ELSE IF(CONDITION)
TRUE
    DISPLAY ALL AVAILABLE FOOD SUPPLEMENT DETAILS
ELSE IF (CONDITION)
TRUE
   DISPLAY ALL AVAILABLE FITNESS EQUIPMENT DETAILS
ELSE IF (CONDITION)
TRUE
    DISPLAY FUNCTIONS TO CALCULATE BILL
ELSE
    DISPLAY "You Inserted an Invalid Option"
END
```

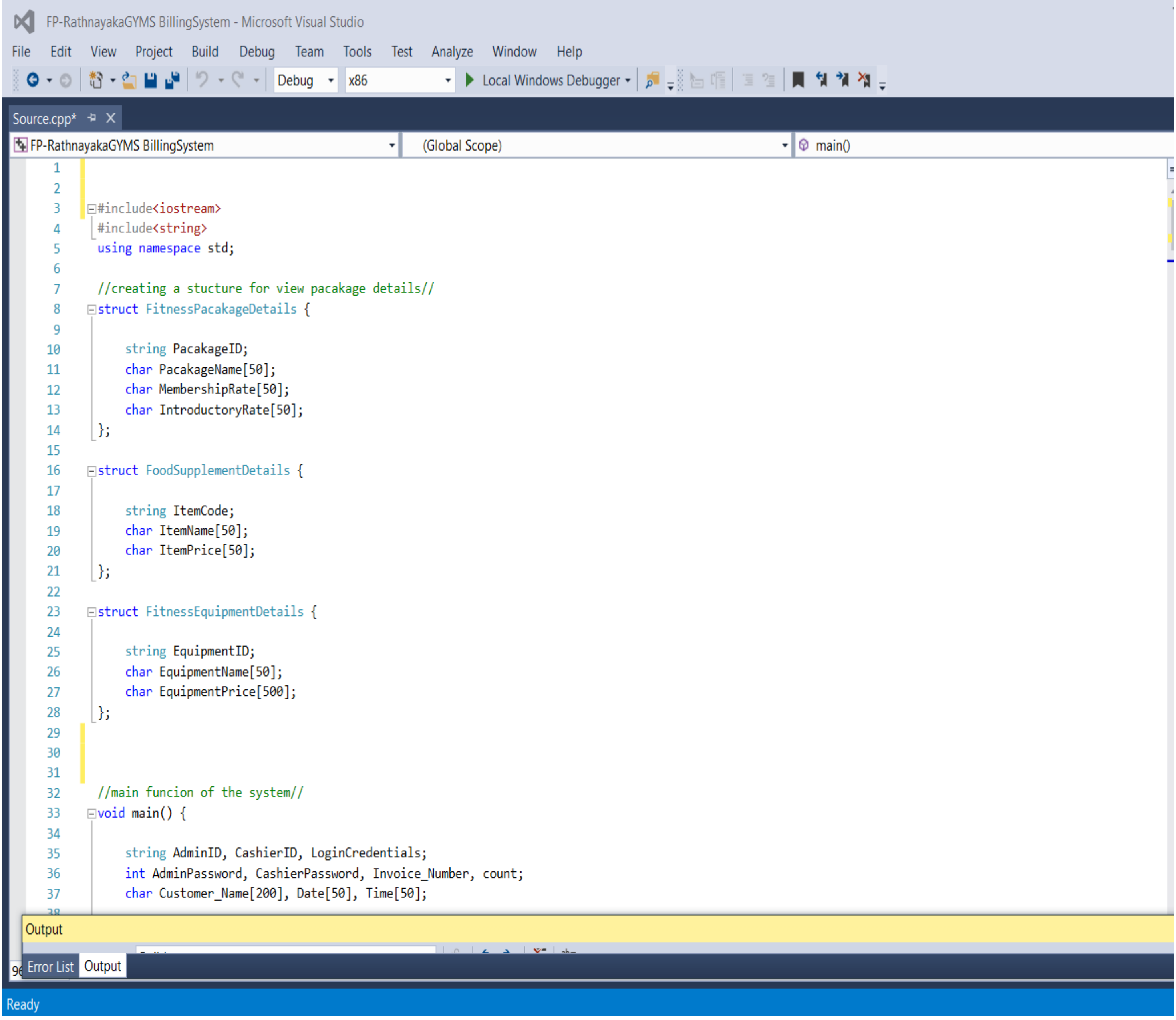
- For Calculation of Bill

```
BEGIN
CONSIDER VARIABLES AS Option,4
INSERT Option 4
    DISPLAY “This is Rathnayaka Gyms Cashiering”
INSERT FOOD SUPPLEMENT ITEM NUMBERS & PRICES
    DISPLAY THE FUNCTION OF SELECTING SUPPLEMENT/S
    DISPLAY THE FUNCTION OF GETTING THE QUANTITY
INSERT FITNESS PACKAGES ID & PRICES
    DISPLAY THE FUNCTION OF SELECTING PACKAGE/S
INSERT FITNESS EQUIPMENTS ID & PRICES
    DISPLAY THE FUNCTION OF SELECTING EQUIPMENT/S
    DISPLAY THE FUNCTION OF GETTING THE QUANTITY
INSERT Invoice_Number
INSERT Customer_Name
INSERT Date
INSERT Time
   DISPLAY RATHNAYAKA GYMS SLOGAN
   DISPLAY “Invoice Number:”
   DISPLAY “Customer’s Name:”
   DISPLAY “Date of issuing this bill is:”
   DISPLAY “Time of issuing this bill is:”
DISPLAY ITEMS ORDERED
DISPLAY PAYMENT SUMMARY
ADD PRICES OF SELECTED ITEMS AND PACKAGES
ASSIGN ANSWER IN Total
DIVIDE Total BY 100*5
ASSIGN ANSWER IN Discount
SUBSTRACT Total BY Discount
ASSIGN ANSWER IN FinalTotal
    DISPLAY “Total is:” + Total
    DISPLAY “Discount for your bill is:” + Discount
    DISPLAY “The Final Total Amount:” +FinalTotal
INSERT Cash
INSERT Payment_Type
   DISPLAY “Cash Amount:”
   DISPLAY “Payment Type:”
SUBSTRACT FinalTotal BY Cash
ASSIGN ANSWER IN Balance
   DISPLAY “Balance is:” + Balance
DISPLAY “END OF THE BILL!” ‘GYMS NAME’ ‘BRNCHES’ “THANK YOU & COME AGAIN”
END
```

Task 3 (System Implementation)

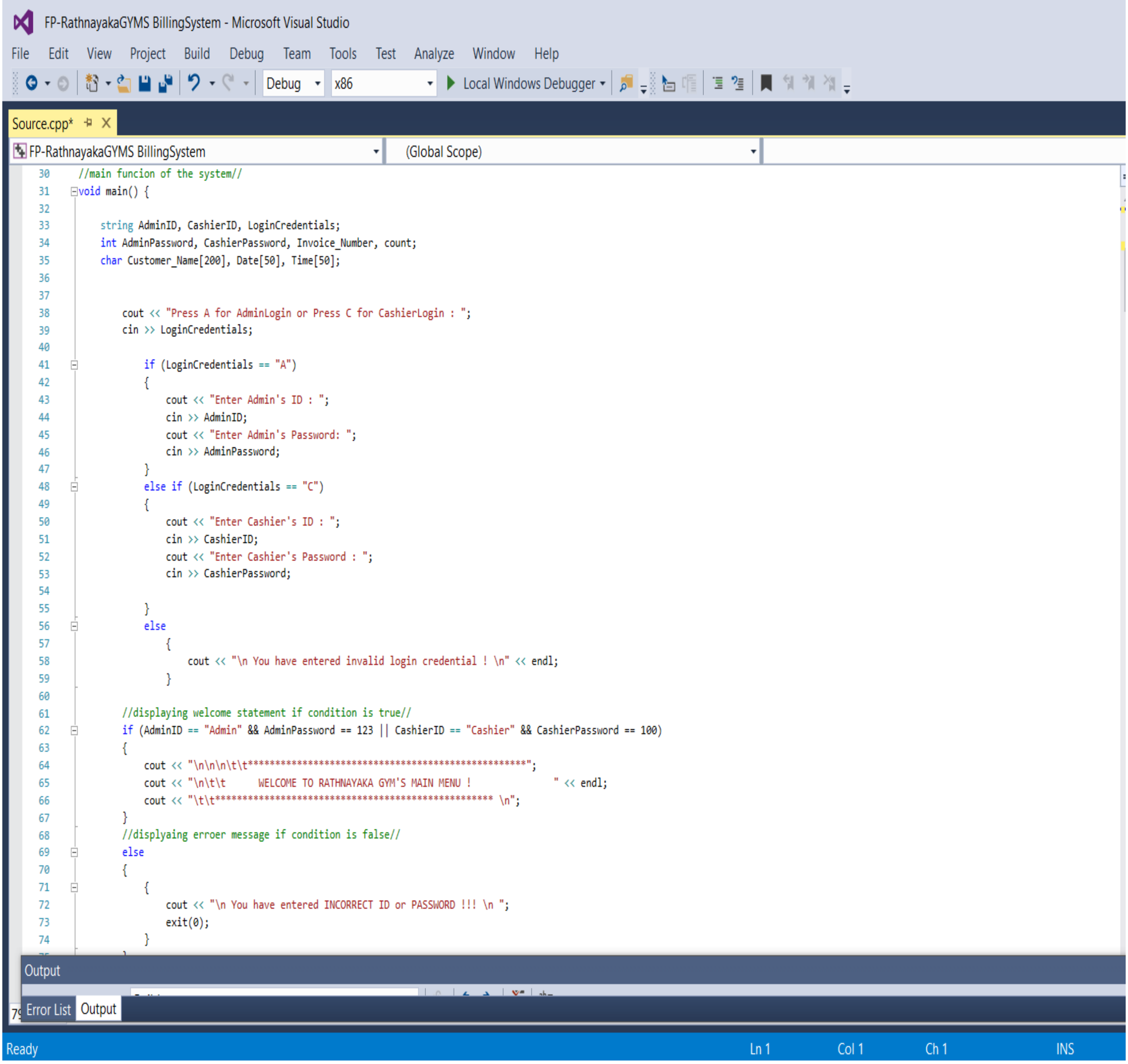
This Rathnayaka Gyms Billing System is implemented using Visual Studio 2015 Professional and as a console application. Thus mainly, this system performs login, view details and calculate bill functions. Below screenshots of the code depicts the core sections in the program.

- In the following screenshot, it shows the header files used at the beginning of the code and then the structures created for each Fitness Package Details, Food Supplement Details and Fitness Equipment Details. And also, the data types used for each are clearly depicted.



```


#include<iostream>
#include<string>
using namespace std;

//creating a stucture for view pacakage details//
struct FitnessPacakageDetails {

    string PacakageID;
    char PacakageName[50];
    char MembershipRate[50];
    char IntroductoryRate[50];
};

struct FoodSupplementDetails {

    string ItemCode;
    char ItemName[50];
    char ItemPrice[50];
};

struct FitnessEquipmentDetails {

    string EquipmentID;
    char EquipmentName[50];
    char EquipmentPrice[500];
};



//main funcion of the system//
void main() {

    string AdminID, CashierID, LoginCredentials;
    int AdminPassword, CashierPassword, Invoice_Number, count;
    char Customer_Name[200], Date[50], Time[50];
```

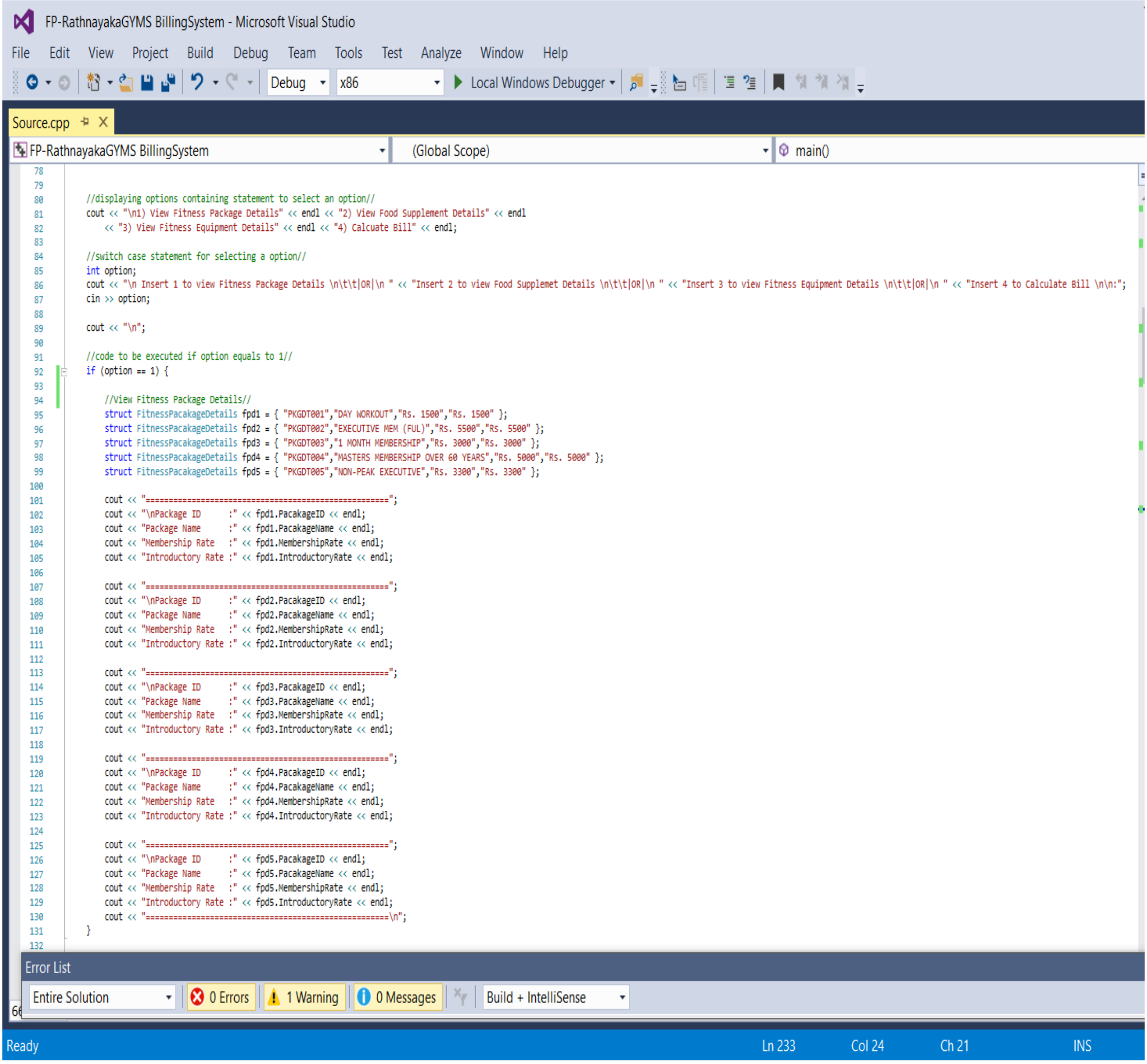
- In the following screenshot, the beginning of the Main Function and using of if…else statements for Login Procedure is clearly displayed. There I have programmed for two Logins for Admin and Cashier.



FP-RathnayakaGYMS BillingSystem - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp*
FP-RathnayakaGYMS BillingSystem
(Global Scope)
Output
Error List
Output
Ready
Ln 1
Col 1
Ch 1
INS

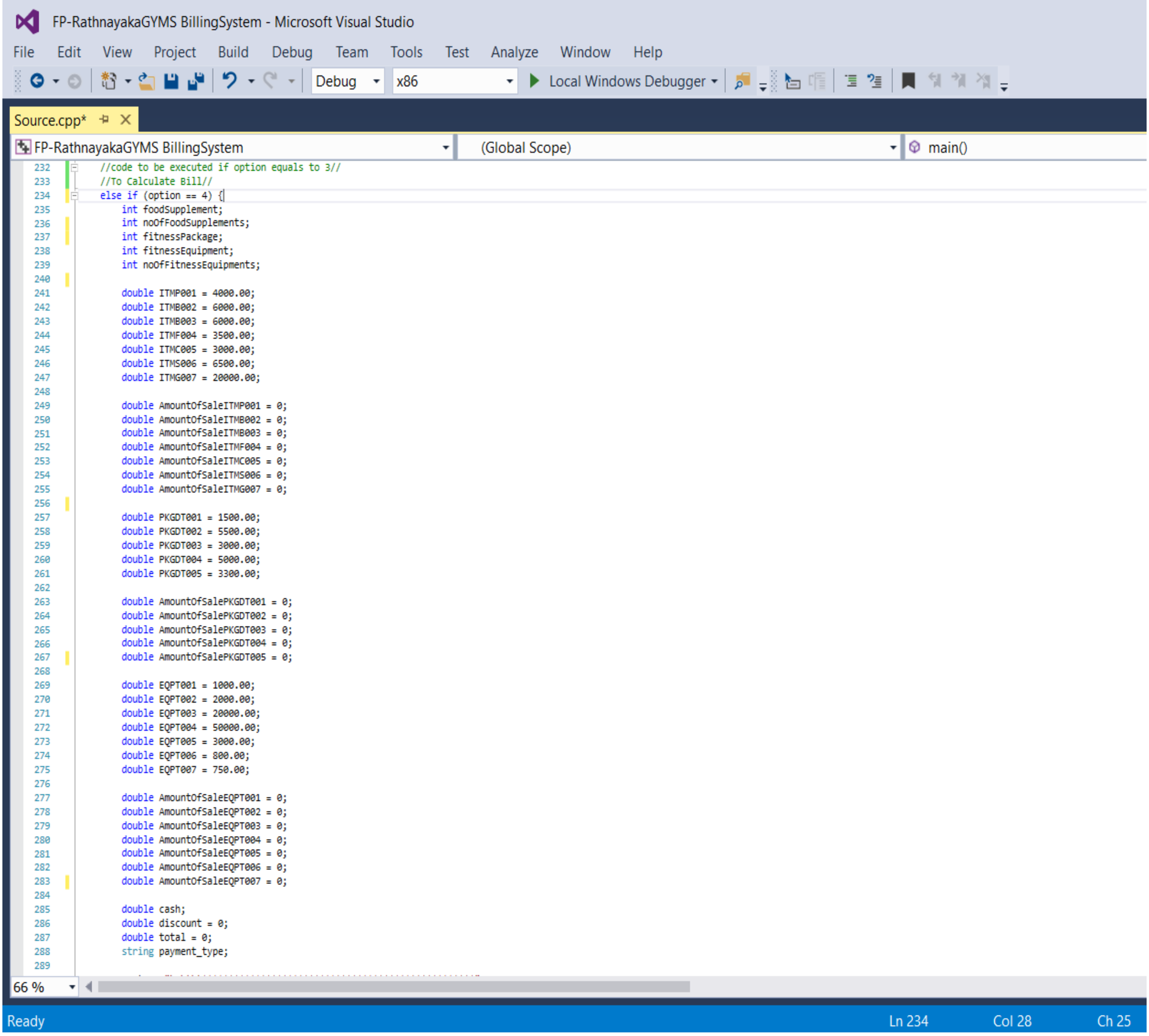
```cpp
//main funcion of the system//
void main() {

    string AdminID, CashierID, LoginCredentials;
    int AdminPassword, CashierPassword, Invoice_Number, count;
    char Customer_Name[200], Date[50], Time[50];


        cout << "Press A for AdminLogin or Press C for CashierLogin : ";
        cin >> LoginCredentials;

            if (LoginCredentials == "A")
            {
                cout << "Enter Admin's ID : ";
                cin >> AdminID;
                cout << "Enter Admin's Password: ";
                cin >> AdminPassword;
            }
            else if (LoginCredentials == "C")
            {
                cout << "Enter Cashier's ID : ";
                cin >> CashierID;
                cout << "Enter Cashier's Password : ";
                cin >> CashierPassword;

            }
            else
                {
                    cout << "\n You have entered invalid login credential ! \n" << endl;
                }

        //displaying welcome statement if condition is true//
        if (AdminID == "Admin" && AdminPassword == 123 || CashierID == "Cashier" && CashierPassword == 100)
        {
            cout << "\n\n\n\t\t*****************************************************";
            cout << "\n\t\t      WELCOME TO RATHNAYAKA GYM'S MAIN MENU !               " << endl;
            cout << "\t\t***************************************************** \n";
        }
        //displyaing erroer message if condition is false//
        else
        {
            {
                cout << "\n You have entered INCORRECT ID or PASSWORD !!! \n ";
                exit(0);
            }
```

- In the following screenshot, at the beginning it shows instructions in the Main Menu and then the options to View Fitness Packages, View Food Supplements, View Fitness Equipment and to Calculate Bill. At the end of the code, I have shown how I inserted data for Viewing Fitness Packages function and inserting data for other functions also continu like this except the calculation of bill.


FP-RathnayakaGYMS BillingSystem - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp
FP-RathnayakaGYMS BillingSystem
(Global Scope)
main()
Error List
Entire Solution
0 Errors
1 Warning
0 Messages
Build + IntelliSense
Ready
Ln 233
Col 24
Ch 21
INS

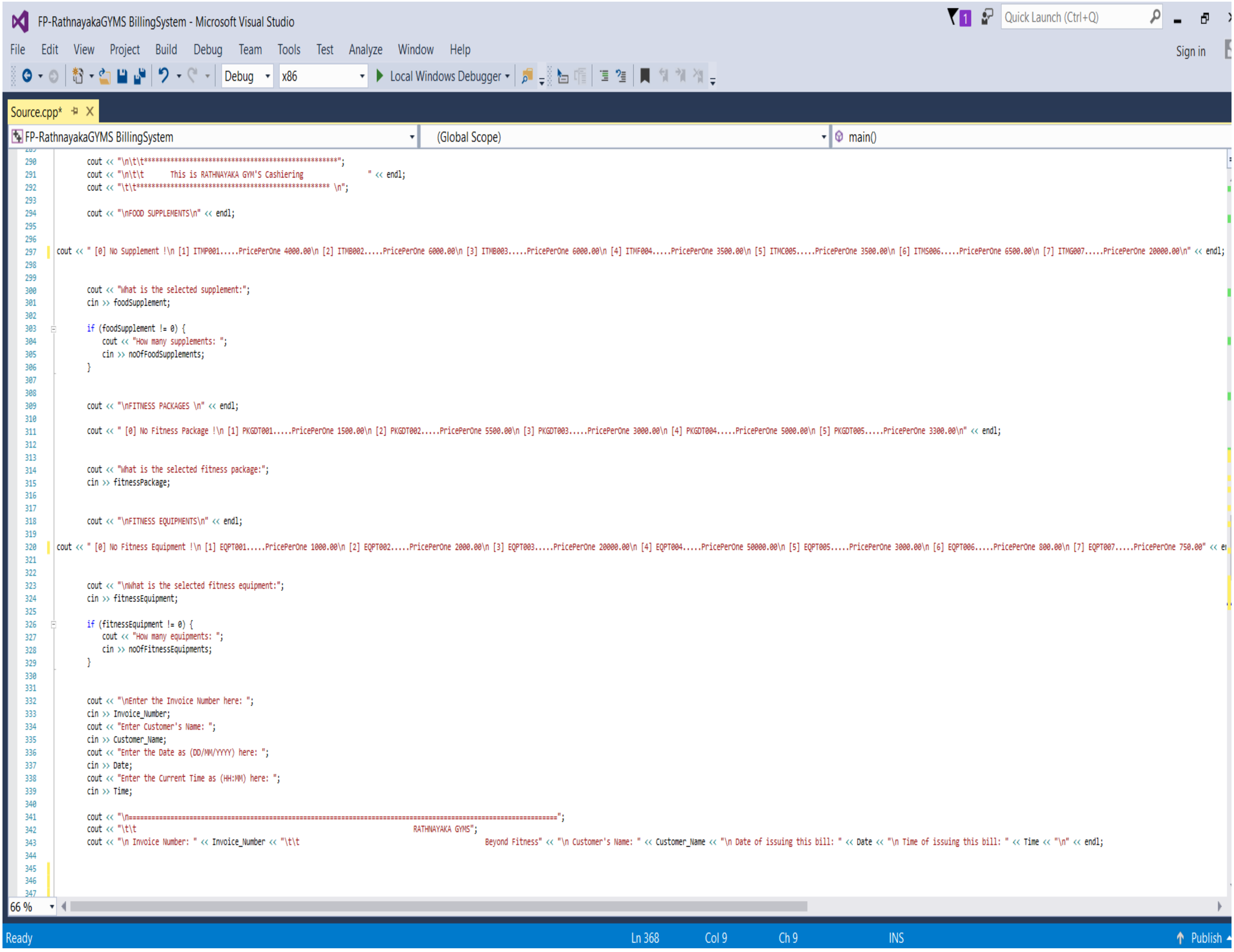
```cpp


    //displaying options containing statement to select an option//
    cout << "\n1) View Fitness Package Details" << endl << "2) View Food Supplement Details" << endl
        << "3) View Fitness Equipment Details" << endl << "4) Calcuate Bill" << endl;

    //switch case statement for selecting a option//
    int option;
    cout << "\n Insert 1 to view Fitness Package Details \n\t\t|OR|\n " << "Insert 2 to view Food Supplemet Details \n\t\t|OR|\n " << "Insert 3 to view Fitness Equipment Details \n\t\t|OR|\n " << "Insert 4 to Calculate Bill \n\n:";
    cin >> option;

    cout << "\n";

    //code to be executed if option equals to 1//
    if (option == 1) {

        //View Fitness Package Details//
        struct FitnessPacakageDetails fpd1 = { "PKGDT001","DAY WORKOUT","Rs. 1500","Rs. 1500" };
        struct FitnessPacakageDetails fpd2 = { "PKGDT002","EXECUTIVE MEM (FUL)","Rs. 5500","Rs. 5500" };
        struct FitnessPacakageDetails fpd3 = { "PKGDT003","1 MONTH MEMBERSHIP","Rs. 3000","Rs. 3000" };
        struct FitnessPacakageDetails fpd4 = { "PKGDT004","MASTERS MEMBERSHIP OVER 60 YEARS","Rs. 5000","Rs. 5000" };
        struct FitnessPacakageDetails fpd5 = { "PKGDT005","NON-PEAK EXECUTIVE","Rs. 3300","Rs. 3300" };

        cout << "=====================================================";
        cout << "\nPackage ID      :" << fpd1.PacakageID << endl;
        cout << "Package Name      :" << fpd1.PacakageName << endl;
        cout << "Membership Rate   :" << fpd1.MembershipRate << endl;
        cout << "Introductory Rate :" << fpd1.IntroductoryRate << endl;

        cout << "=====================================================";
        cout << "\nPackage ID      :" << fpd2.PacakageID << endl;
        cout << "Package Name      :" << fpd2.PacakageName << endl;
        cout << "Membership Rate   :" << fpd2.MembershipRate << endl;
        cout << "Introductory Rate :" << fpd2.IntroductoryRate << endl;

        cout << "=====================================================";
        cout << "\nPackage ID      :" << fpd3.PacakageID << endl;
        cout << "Package Name      :" << fpd3.PacakageName << endl;
        cout << "Membership Rate   :" << fpd3.MembershipRate << endl;
        cout << "Introductory Rate :" << fpd3.IntroductoryRate << endl;

        cout << "=====================================================";
        cout << "\nPackage ID      :" << fpd4.PacakageID << endl;
        cout << "Package Name      :" << fpd4.PacakageName << endl;
        cout << "Membership Rate   :" << fpd4.MembershipRate << endl;
        cout << "Introductory Rate :" << fpd4.IntroductoryRate << endl;

        cout << "=====================================================";
        cout << "\nPackage ID      :" << fpd5.PacakageID << endl;
        cout << "Package Name      :" << fpd5.PacakageName << endl;
        cout << "Membership Rate   :" << fpd5.MembershipRate << endl;
        cout << "Introductory Rate :" << fpd5.IntroductoryRate << endl;
        cout << "===================================================\n";
    }

```

- In the following screenshot, I have declared all the data types I used to code all the functions inside the Calculate Bill function. The whole Bill Calculation codes are included inside the option 4.



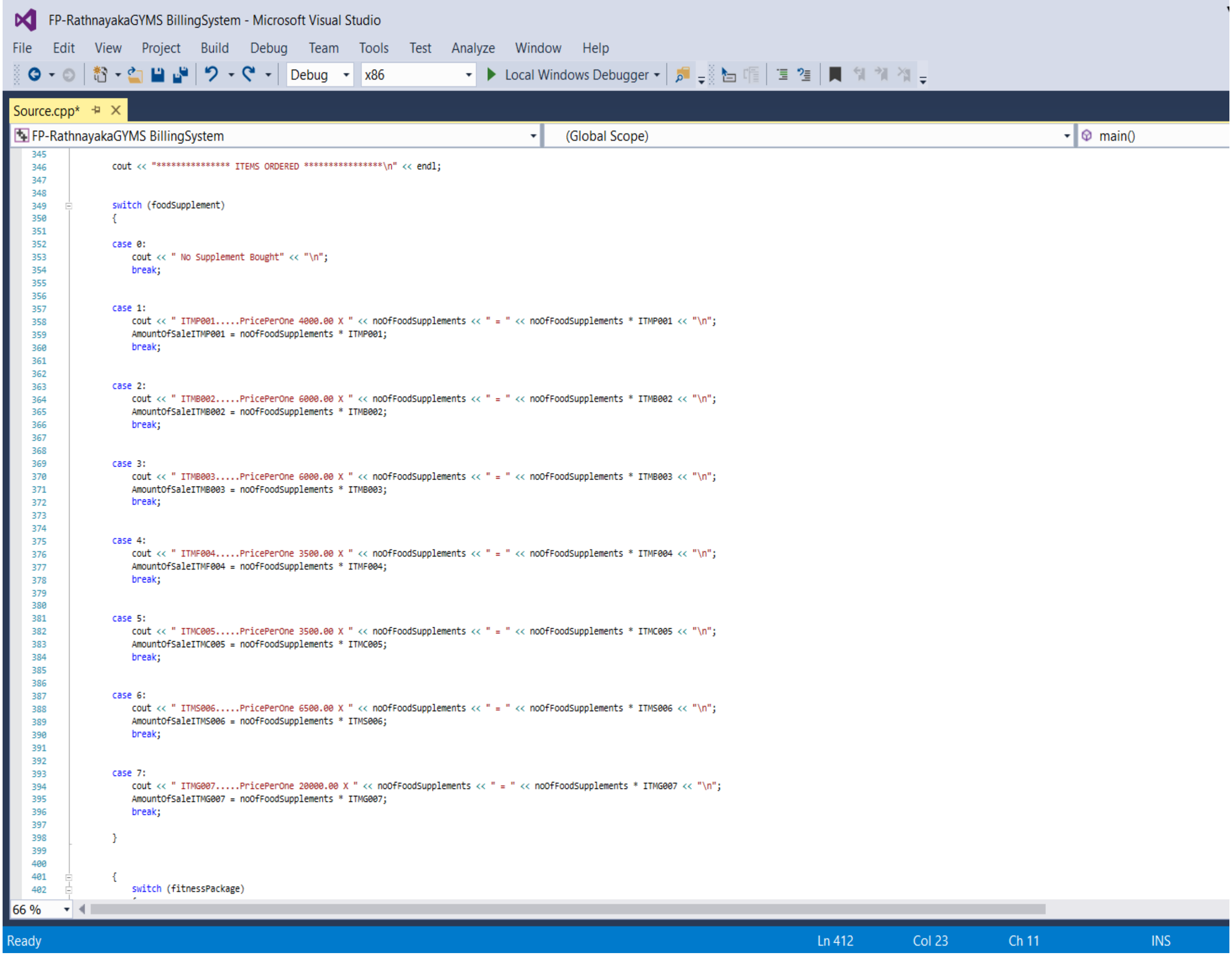
```cpp
//code to be executed if option equals to 3//
//To Calculate Bill//
else if (option == 4) {
    int foodSupplement;
    int noOfFoodSupplements;
    int fitnessPackage;
    int fitnessEquipment;
    int noOfFitnessEquipments;

    double ITMP001 = 4000.00;
    double ITMB002 = 6000.00;
    double ITMB003 = 6000.00;
    double ITMF004 = 3500.00;
    double ITMC005 = 3000.00;
    double ITMS006 = 6500.00;
    double ITMG007 = 20000.00;

    double AmountOfSaleITMP001 = 0;
    double AmountOfSaleITMB002 = 0;
    double AmountOfSaleITMB003 = 0;
    double AmountOfSaleITMF004 = 0;
    double AmountOfSaleITMC005 = 0;
    double AmountOfSaleITMS006 = 0;
    double AmountOfSaleITMG007 = 0;

    double PKGDT001 = 1500.00;
    double PKGDT002 = 5500.00;
    double PKGDT003 = 3000.00;
    double PKGDT004 = 5000.00;
    double PKGDT005 = 3300.00;

    double AmountOfSalePKGDT001 = 0;
    double AmountOfSalePKGDT002 = 0;
    double AmountOfSalePKGDT003 = 0;
    double AmountOfSalePKGDT004 = 0;
    double AmountOfSalePKGDT005 = 0;

    double EQPT001 = 1000.00;
    double EQPT002 = 2000.00;
    double EQPT003 = 20000.00;
    double EQPT004 = 50000.00;
    double EQPT005 = 3000.00;
    double EQPT006 = 800.00;
    double EQPT007 = 750.00;

    double AmountOfSaleEQPT001 = 0;
    double AmountOfSaleEQPT002 = 0;
    double AmountOfSaleEQPT003 = 0;
    double AmountOfSaleEQPT004 = 0;
    double AmountOfSaleEQPT005 = 0;
    double AmountOfSaleEQPT006 = 0;
    double AmountOfSaleEQPT007 = 0;

    double cash;
    double discount = 0;
    double total = 0;
    string payment_type;

```

- In the below screenshot, it shows the how I have code to get the food supplement item number along with their prices, fitness package id along with the prices and fitness equipment id and their prices. Further, the occasions where it asks for selecting particular supplements, packages, equipment and also entering quantity to proceed the bill. Moreover, if no supplement, package or equipment entering of 0(zero) for continuing of bill is also done there.


FP-RathnayakaGYMS BillingSystem - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Sign in
Debug
x86
Local Windows Debugger
Source.cpp*
FP-RathnayakaGYMS BillingSystem
(Global Scope)
main()
66 %
Ready
Ln 368
Col 9
Ch 9
INS
Publish

```cpp
        cout << "\n\t\t**************************************************";
        cout << "\n\t\t      This is RATHNAYAKA GYM'S Cashiering                   " << endl;
        cout << "\t\t************************************************** \n";

        cout << "\nFOOD SUPPLEMENTS\n" << endl;


cout << " [0] No Supplement !\n [1] ITMP001.....PricePerOne 4000.00\n [2] ITMB002.....PricePerOne 6000.00\n [3] ITMB003.....PricePerOne 6000.00\n [4] ITMF004.....PricePerOne 3500.00\n [5] ITMC005.....PricePerOne 3500.00\n [6] ITMS006.....PricePerOne 6500.00\n [7] ITMG007.....PricePerOne 20000.00\n" << endl;


        cout << "What is the selected supplement:";
        cin >> foodSupplement;

        if (foodSupplement != 0) {
            cout << "How many supplements: ";
            cin >> noOfFoodSupplements;
        }


        cout << "\nFITNESS PACKAGES \n" << endl;

        cout << " [0] No Fitness Package !\n [1] PKGDT001.....PricePerOne 1500.00\n [2] PKGDT002.....PricePerOne 5500.00\n [3] PKGDT003.....PricePerOne 3000.00\n [4] PKGDT004.....PricePerOne 5000.00\n [5] PKGDT005.....PricePerOne 3300.00\n" << endl;


        cout << "What is the selected fitness package:";
        cin >> fitnessPackage;


        cout << "\nFITNESS EQUIPMENTS\n" << endl;

cout << " [0] No Fitness Equipment !\n [1] EQPT001.....PricePerOne 1000.00\n [2] EQPT002.....PricePerOne 2000.00\n [3] EQPT003.....PricePerOne 20000.00\n [4] EQPT004.....PricePerOne 50000.00\n [5] EQPT005.....PricePerOne 3000.00\n [6] EQPT006.....PricePerOne 800.00\n [7] EQPT007.....PricePerOne 750.00" << en


        cout << "\nWhat is the selected fitness equipment:";
        cin >> fitnessEquipment;

        if (fitnessEquipment != 0) {
            cout << "How many equipments: ";
            cin >> noOfFitnessEquipments;
        }


        cout << "\nEnter the Invoice Number here: ";
        cin >> Invoice_Number;
        cout << "Enter Customer's Name: ";
        cin >> Customer_Name;
        cout << "Enter the Date as (DD/MM/YYYY) here: ";
        cin >> Date;
        cout << "Enter the Current Time as (HH:MM) here: ";
        cin >> Time;

        cout << "\n====================================================================================================";
        cout << "\t\t                                                              RATHNAYAKA GYMS";
        cout << "\n Invoice Number: " << Invoice_Number << "\t\t                                                  Beyond Fitness" << "\n Customer's Name: " << Customer_Name << "\n Date of issuing this bill: " << Date << "\n Time of issuing this bill: " << Time << "\n" << endl;
```

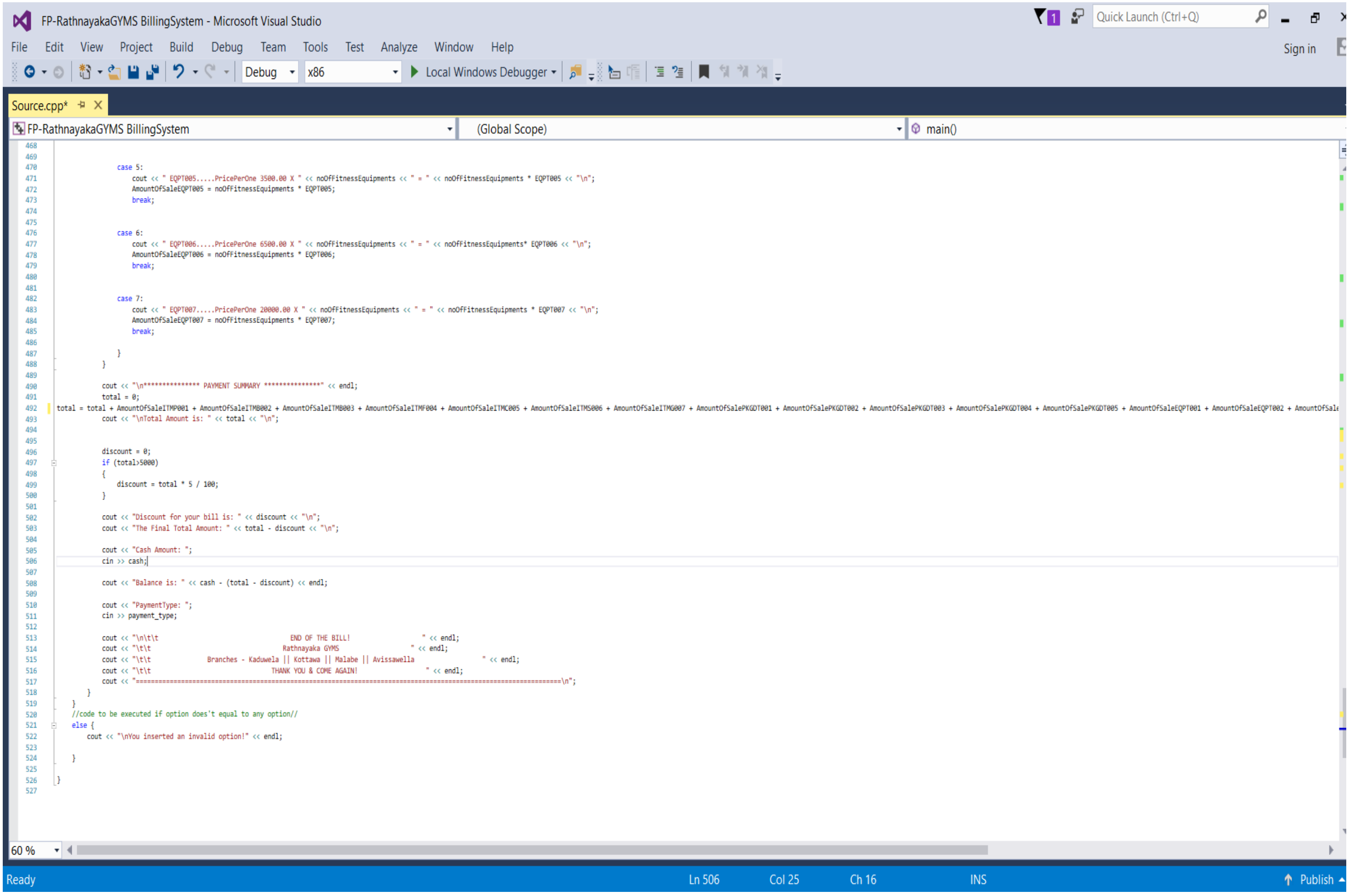
- In the screenshot which display below, it shows how I have used switch case statements for selecting a supplement and its going to added in the bill.



FP-RathnayakaGYMS BillingSystem - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Source.cpp*
FP-RathnayakaGYMS BillingSystem
(Global Scope)
main()
66 %
Ready
Ln 412
Col 23
Ch 11
INS

```
        cout << "*************** ITEMS ORDERED ****************\n" << endl;


        switch (foodSupplement)
        {

        case 0:
            cout << " No Supplement Bought" << "\n";
            break;


        case 1:
            cout << " ITMP001.....PricePerOne 4000.00 X " << noOfFoodSupplements << " = " << noOfFoodSupplements * ITMP001 << "\n";
            AmountOfSaleITMP001 = noOfFoodSupplements * ITMP001;
            break;


        case 2:
            cout << " ITMB002.....PricePerOne 6000.00 X " << noOfFoodSupplements << " = " << noOfFoodSupplements * ITMB002 << "\n";
            AmountOfSaleITMB002 = noOfFoodSupplements * ITMB002;
            break;


        case 3:
            cout << " ITMB003.....PricePerOne 6000.00 X " << noOfFoodSupplements << " = " << noOfFoodSupplements * ITMB003 << "\n";
            AmountOfSaleITMB003 = noOfFoodSupplements * ITMB003;
            break;


        case 4:
            cout << " ITMF004.....PricePerOne 3500.00 X " << noOfFoodSupplements << " = " << noOfFoodSupplements * ITMF004 << "\n";
            AmountOfSaleITMF004 = noOfFoodSupplements * ITMF004;
            break;


        case 5:
            cout << " ITMC005.....PricePerOne 3500.00 X " << noOfFoodSupplements << " = " << noOfFoodSupplements * ITMC005 << "\n";
            AmountOfSaleITMC005 = noOfFoodSupplements * ITMC005;
            break;


        case 6:
            cout << " ITMS006.....PricePerOne 6500.00 X " << noOfFoodSupplements << " = " << noOfFoodSupplements * ITMS006 << "\n";
            AmountOfSaleITMS006 = noOfFoodSupplements * ITMS006;
            break;


        case 7:
            cout << " ITMG007.....PricePerOne 20000.00 X " << noOfFoodSupplements << " = " << noOfFoodSupplements * ITMG007 << "\n";
            AmountOfSaleITMG007 = noOfFoodSupplements * ITMG007;
            break;

        }


        {
            switch (fitnessPackage)
```

- The below screenshot is attached to depict the ending of the code. It includes how the total is calculated, how the discount is calculated, how cash amount entered, how balance is returned and how they are coded to get the output. Finally, the latter part of the screenshot displays the stuffs that are going to be displayed at the end of the bill.


FP-RathnayakaGYMS BillingSystem - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Sign in
Debug
x86
Local Windows Debugger
Source.cpp*
FP-RathnayakaGYMS BillingSystem
(Global Scope)
main()
60 %
Ready
Ln 506
Col 25
Ch 16
INS
Publish

```


            case 5:
                cout << " EQPT005.....PricePerOne 3500.00 X " << noOfFitnessEquipments << " = " << noOfFitnessEquipments * EQPT005 << "\n";
                AmountOfSaleEQPT005 = noOfFitnessEquipments * EQPT005;
                break;


            case 6:
                cout << " EQPT006.....PricePerOne 6500.00 X " << noOfFitnessEquipments << " = " << noOfFitnessEquipments* EQPT006 << "\n";
                AmountOfSaleEQPT006 = noOfFitnessEquipments * EQPT006;
                break;


            case 7:
                cout << " EQPT007.....PricePerOne 20000.00 X " << noOfFitnessEquipments << " = " << noOfFitnessEquipments * EQPT007 << "\n";
                AmountOfSaleEQPT007 = noOfFitnessEquipments * EQPT007;
                break;

            }
        }

        cout << "\n*************** PAYMENT SUMMARY ***************" << endl;
        total = 0;
total = total + AmountOfSaleITMP001 + AmountOfSaleITMB002 + AmountOfSaleITMB003 + AmountOfSaleITMF004 + AmountOfSaleITMC005 + AmountOfSaleITMS006 + AmountOfSaleITMG007 + AmountOfSalePKGDT001 + AmountOfSalePKGDT002 + AmountOfSalePKGDT003 + AmountOfSalePKGDT004 + AmountOfSalePKGDT005 + AmountOfSaleEQPT001 + AmountOfSaleEQPT002 + AmountOfSale
        cout << "\nTotal Amount is: " << total << "\n";


        discount = 0;
        if (total>5000)
        {
            discount = total * 5 / 100;
        }

        cout << "Discount for your bill is: " << discount << "\n";
        cout << "The Final Total Amount: " << total - discount << "\n";

        cout << "Cash Amount: ";
        cin >> cash;

        cout << "Balance is: " << cash - (total - discount) << endl;

        cout << "PaymentType: ";
        cin >> payment_type;

        cout << "\n\t\t                                END OF THE BILL!                  " << endl;
        cout << "\t\t                               Rathnayaka GYMS                  " << endl;
        cout << "\t\t              Branches - Kaduwela || Kottawa || Malabe || Avissawella                 " << endl;
        cout << "\t\t                            THANK YOU & COME AGAIN!                 " << endl;
        cout << "==================================================================================================\n";
    }
  }
  //code to be executed if option does't equal to any option//
  else {
      cout << "\nYou inserted an invalid option!" << endl;

  }

}

```

Task 4 (System Testing & Documentation)

4.1) Selected Testing Technique

Black Box Testing

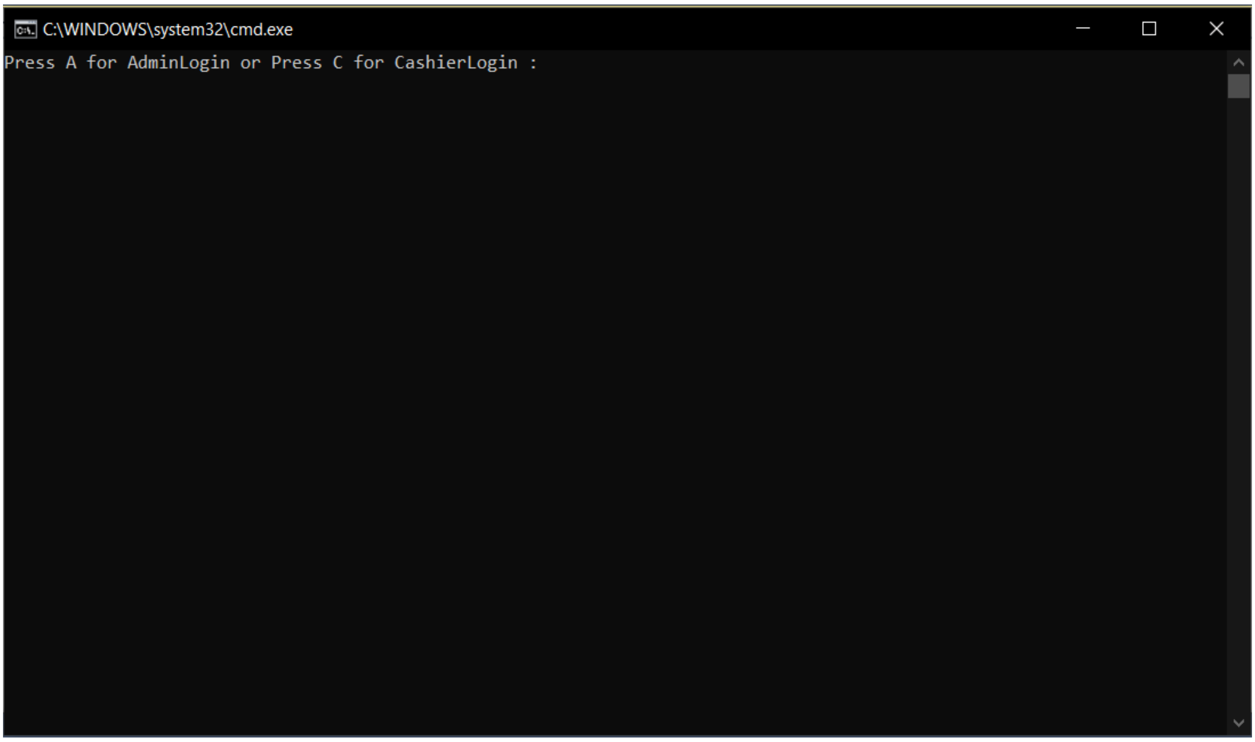
As White Box Testing more costly than the Black Box Testing, Black Box Testing is efficient to use here. And few reasons for choosing Black Box Testing are It checks the functions of the Billing System and more importantly it does not check the code. It checks the outputs based on the inputs provided. Therefore, it can be referred as external software testing as well. Further, This Black Box testing can begin based on the requirement specification document and the time consuming is less. This can be functional, non-functional or Regression testing.

4.2) Test Plan

TEST CASE NUMBER	TEST CASE NAME	STEPS	EXPECTED RESULT
TC1	Checking whether login credentials are working properly	Debug and check	Should display enter login credentials option
TC1.1	Checking correctness of the login credentials	Enter correct login credentials (Insert A or C)	Should display “Enter Admin's ID:”
TC1.2	Checking correctness of the login credentials	Enter incorrect login credential (any letter except A & C)	Should display “You have entered invalid login credential! You have entered INCORRECT ID or PASSWORD!!! Press any key to continue . . .”
TC2	Login	Enter relevant correct ID & Password	Should display the main menu with the functions in the main menu
TC2.1	Login	Enter incorrect ID or Password	Should display “You have entered INCORRECT ID or PASSWORD!!! Press any key to continue . . .”
TC3	Main Menu	Insert correct login credential, ID & Password	Automatically, the Main Menu should be displayed with the functions and options
TC3.1	Main Menu	Insert 1 to view all the fitness package details	Should display all the fitness packages with relevant data assigned for each Package ID, Package Name, Membership Rate, Introductory Rate
TC3.2	Main Menu	Insert 2 to view all the food supplement details	Should display all the food supplement details with relevant data assigned for each Item Code, Item Name, Item Price
TC3.3	Main Menu	Insert 3 to view all the fitness equipment details	Should display all the fitness equipment details with relevant data assigned for each Equipment Code, Equipment Name, Equipment Price
TC3.4	Main Menu	Insert 4 to Calculate Bill	Should display “This is RATHNAYAKA GYM'S Cashiering” and should work the functions one by one which are assigned for calculating bill

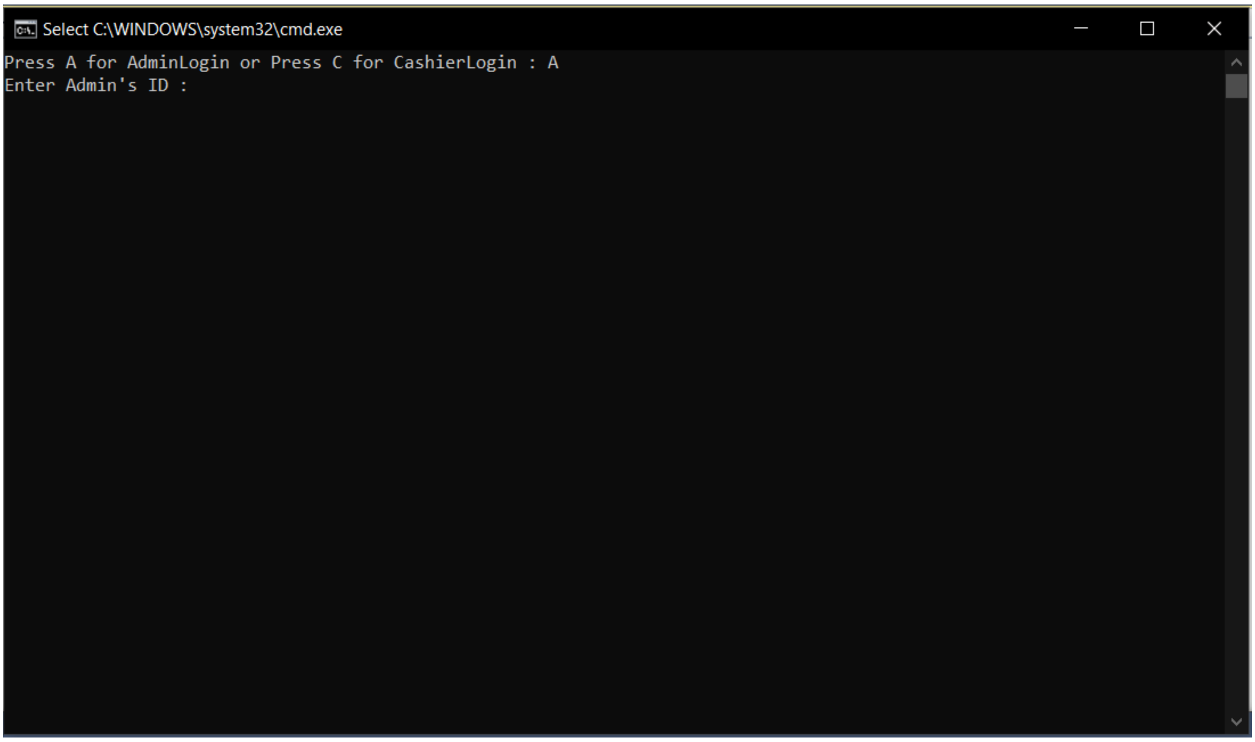
TC3.5	Main Menu	Insert a number except 1,2,3 & 4	Should display "You inserted an invalid option! Press any key to continue . . ."
TC4	Calculate Bill	Insert number 4 option in the Main Menu	Should Display "This is RATHNAYAKA GYM'S Cashiering" and the food supplements with item code and Price
TC4.1	Calculate Bill (Food Supplements)	Enter number relevant to the supplement ID of the bought supplement, enter & insert the quantity using a number	Should display the fitness packages with package ID and Price and should display "What is the selected fitness package:" function
TC4.2	Calculate Bill (Fitness Packages)	Enter number relevant to the package ID of the package selected & enter	Should display the fitness equipment with equipment ID and Price and should display "What is the selected fitness equipment:" function
TC4.3	Calculate Bill (Fitness Equipment)	Enter number relevant to the equipment ID of the bought equipment, enter & insert the quantity using a number	Should display the "Enter the Invoice Number here:" function
TC4.4	Calculate Bill	Enter any number for invoice number, enter customer name like this (ChamathShyamal) without any space, enter the current date like this (19/04/2021) & enter the current time like (1.45 or 22.16)	Should display Invoice number, Customer's name, Date & Time in order with Rathnayaka Gyms slogan at the right bottom & then the Items Ordered
TC4.5	Calculate Bill (Payment Summary)	--------	Should display Total Amount, Discount, The Final Total Amount after deducting the discount from the Total & Cash Amount that the customer going to pay
TC4.5.1	Calculate Bill (Payment Summary)	Enter the amount paid by the customer Infront of Cash Amount, enter and then insert the payment type	Should display Balance amount that should return to customer and whether customer paid using a card or using cash
TC4.5.2	Ending of the Bill	After inserting the payment type, press 'Enter'	Should display "END OF THE BILL! Gym Name (Rathnayaka Gyms), branches and THANK YOU & COME AGAIN! Press any key to continue . . ."

4.3) Test Cases

TEST CASE NUMBER	TC1
TEST CASE NAME	Checking whether login credentials are working properly
SCENARIO	The purpose of conducting this test case is to check that login credentials work properly
EXPECTED RESULT	Should display enter login credentials option
ACTUAL RESULT	`Press A for AdminLogin or Press C for CashierLogin :`
SCREENSHOT	 `Press A for AdminLogin or Press C for CashierLogin :`
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC1.1
TEST CASE NAME	Checking correctness of the login credentials
SCENARIO	The purpose of conducting this test case is to check the correctness of the login credentials
EXPECTED RESULT	Should display “Enter Admin's ID:”

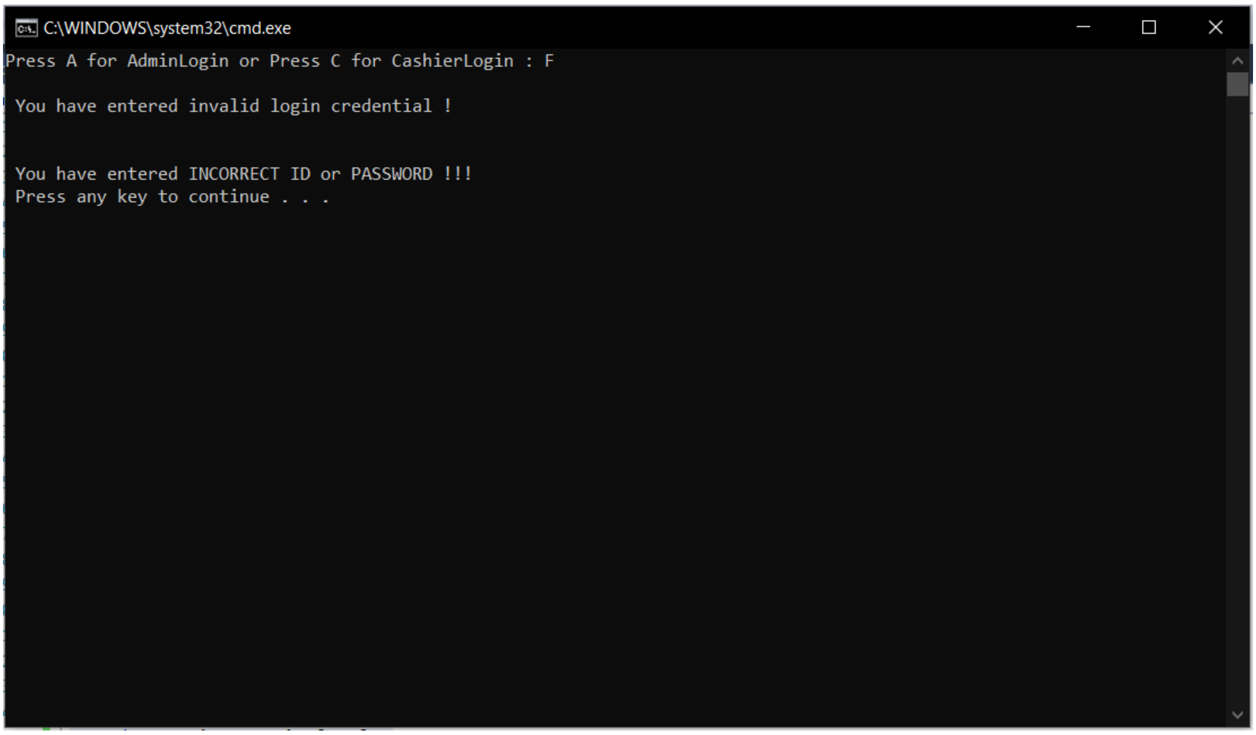
ACTUAL RESULT	`Press A for AdminLogin or Press C for CashierLogin : A`
	`Enter Admin's ID :`

SCREENSHOT	
	
	`Press A for AdminLogin or Press C for CashierLogin : A`
	`Enter Admin's ID :`

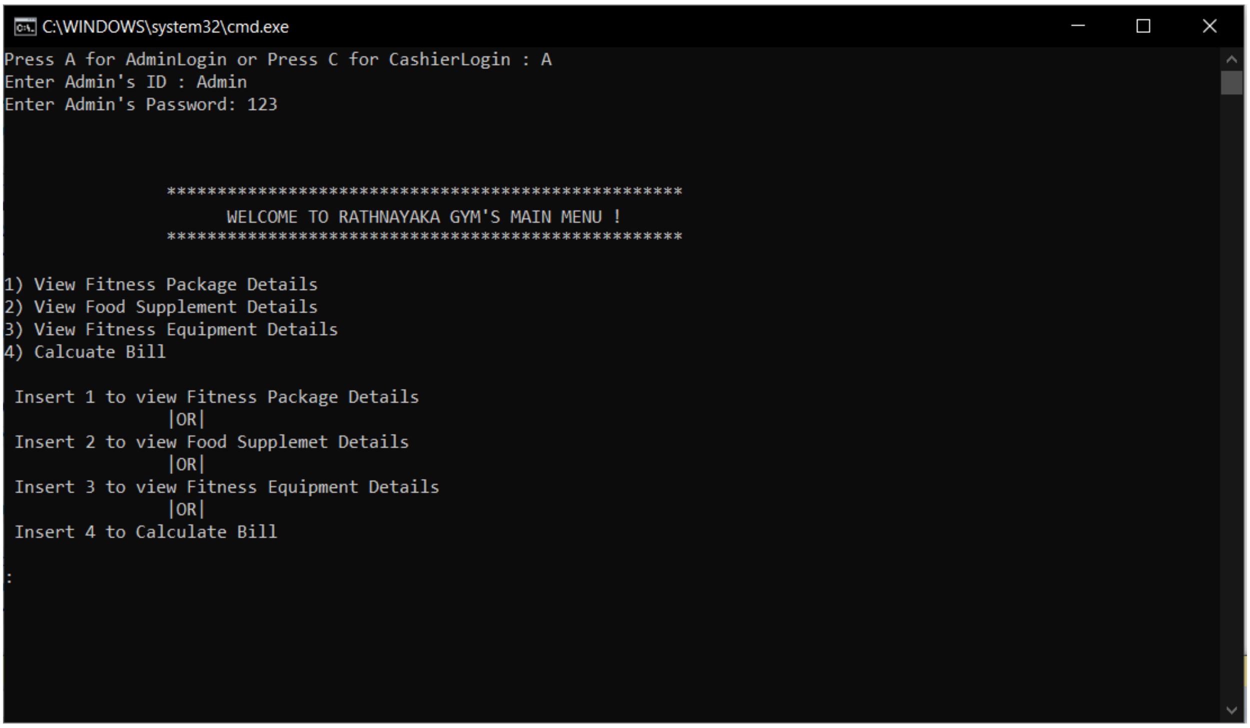
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC1.2
TEST CASE NAME	Checking correctness of the login credentials
SCENARIO	The purpose of conducting this test case is to check whether it is working even when entered incorrect login credential or not
EXPECTED RESULT	Should display “You have entered invalid login credential! You have entered INCORRECT ID or PASSWORD!!! Press any key to continue . . .”

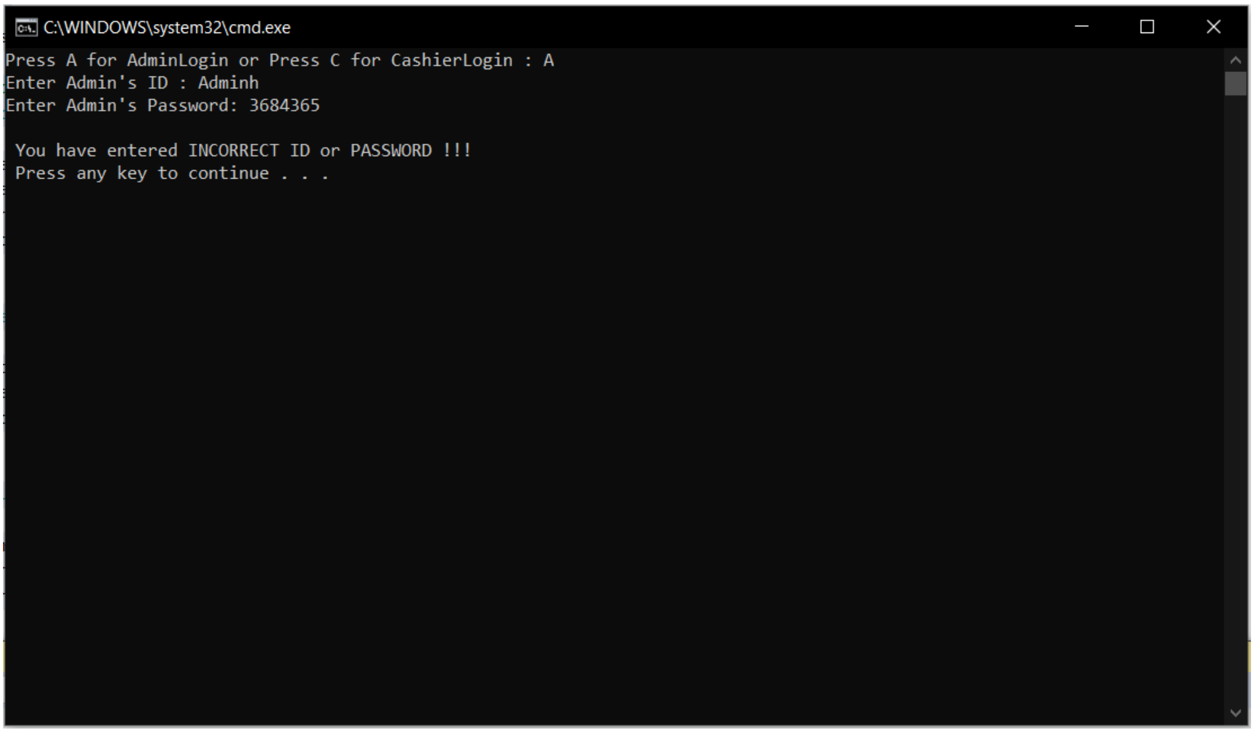
ACTUAL RESULT	`You have entered invalid login credential !`
	`You have entered INCORRECT ID or PASSWORD !!!`
	`Press any key to continue . . .`

SCREENSHOT	
	
	`Press A for AdminLogin or Press C for CashierLogin : F`
	`You have entered invalid login credential !`
	`You have entered INCORRECT ID or PASSWORD !!!`
	`Press any key to continue . . .`

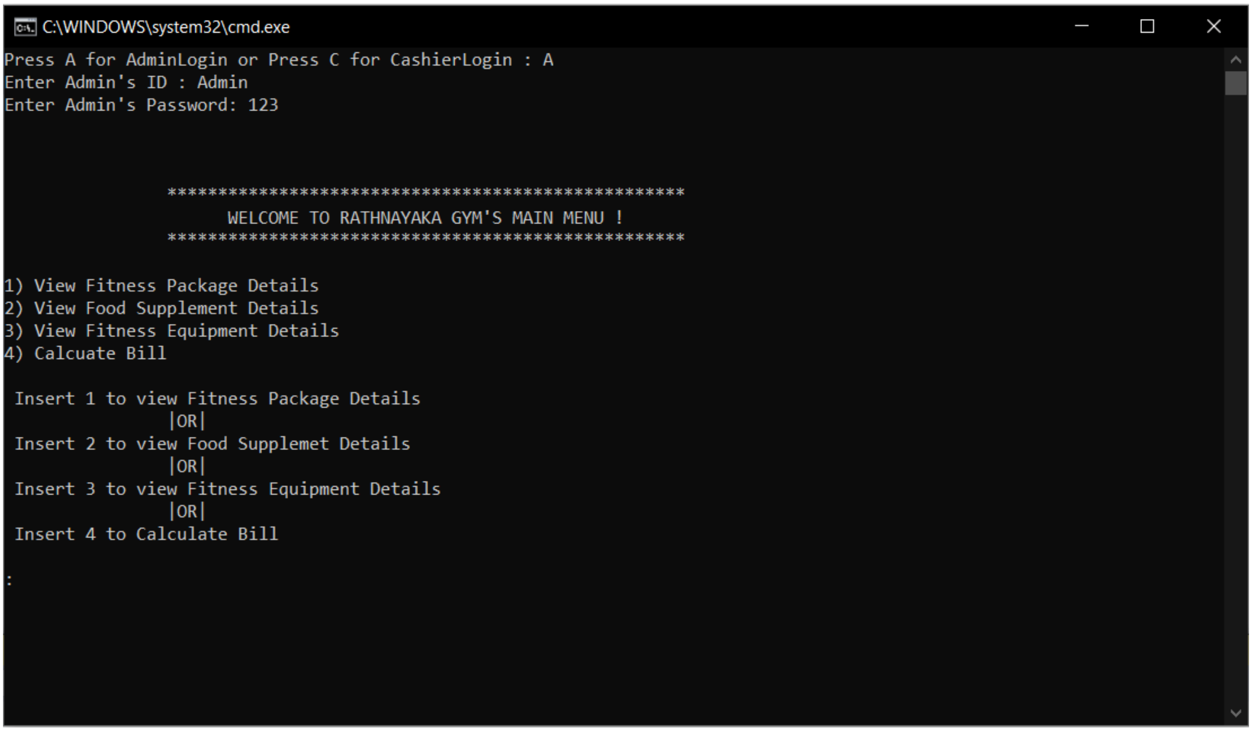
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC2
TEST CASE NAME	Login
SCENARIO	The purpose of conducting this test case is to check whether the program works well when correct inputs (ID & Password) given
EXPECTED RESULT	Should display the main menu with the functions in the main menu
ACTUAL RESULT	Display the main menu and functions in it well
SCREENSHOT	
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

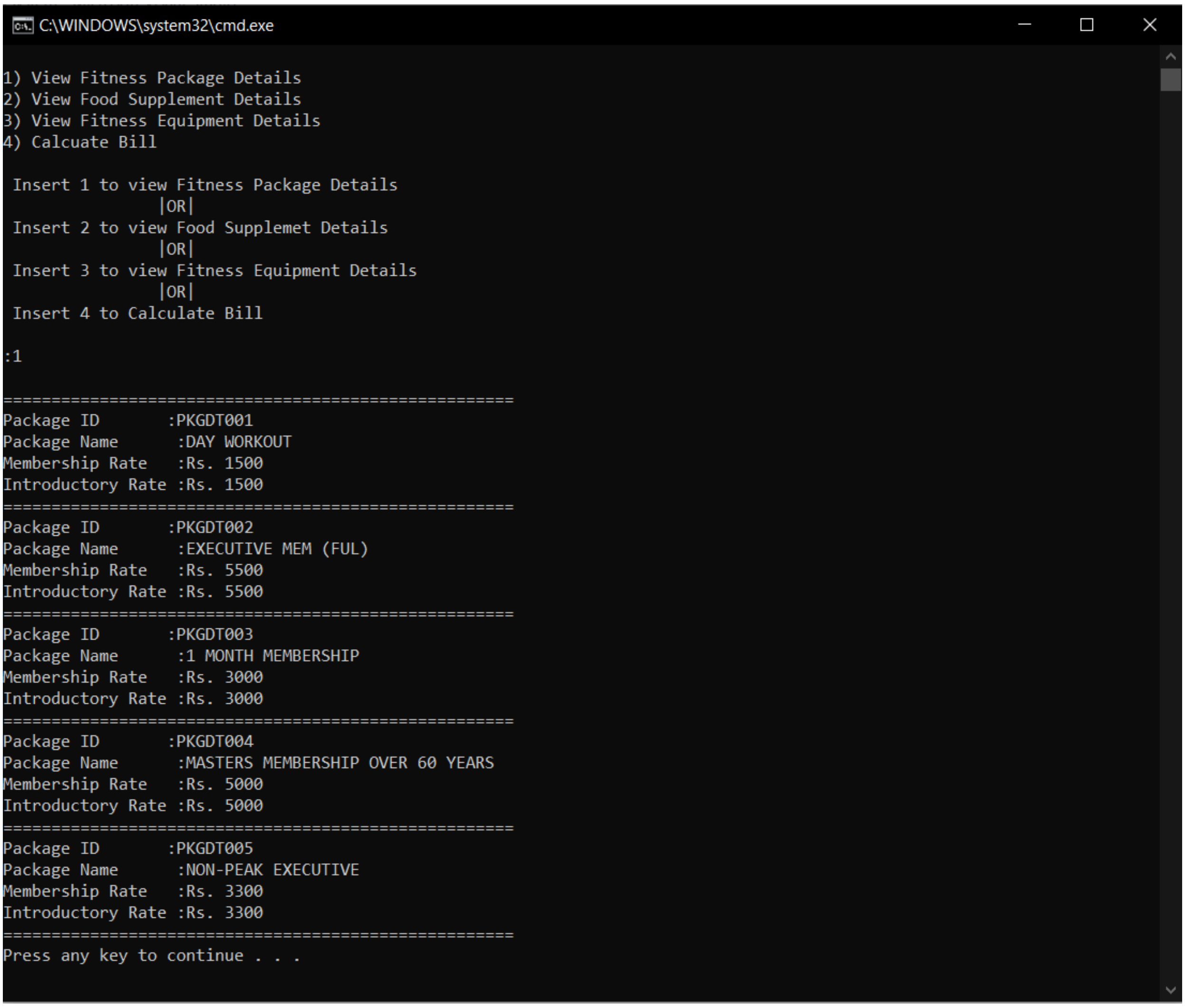
```
Press A for AdminLogin or Press C for CashierLogin : A
Enter Admin's ID : Admin
Enter Admin's Password: 123



                **************************************************
                      WELCOME TO RATHNAYAKA GYM'S MAIN MENU !
                **************************************************

1) View Fitness Package Details
2) View Food Supplement Details
3) View Fitness Equipment Details
4) Calcuate Bill

 Insert 1 to view Fitness Package Details
                |OR|
 Insert 2 to view Food Supplemet Details
                |OR|
 Insert 3 to view Fitness Equipment Details
                |OR|
 Insert 4 to Calculate Bill

:
```

TEST CASE NUMBER	TC2.1
TEST CASE NAME	Login
SCENARIO	The purpose of conducting this test case is to check that the program displays the main menu even after entering wrong ID & Password
EXPECTED RESULT	Should display “You have entered INCORRECT ID or PASSWORD!!! Press any key to continue . . .”
ACTUAL RESULT	You have entered INCORRECT ID or PASSWORD !!! Press any key to continue . . .
SCREENSHOT	 Press A for AdminLogin or Press C for CashierLogin : A Enter Admin's ID : Adminh Enter Admin's Password: 3684365 You have entered INCORRECT ID or PASSWORD !!! Press any key to continue . . .
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC3
TEST CASE NAME	Main Menu
SCENARIO	The purpose of conducting this test case is to check whether the Main Menu displays after entering correct inputs
EXPECTED RESULT	Automatically, the Main Menu should be displayed with the functions and options
ACTUAL RESULT	The Main Menu displays well with the functions in it

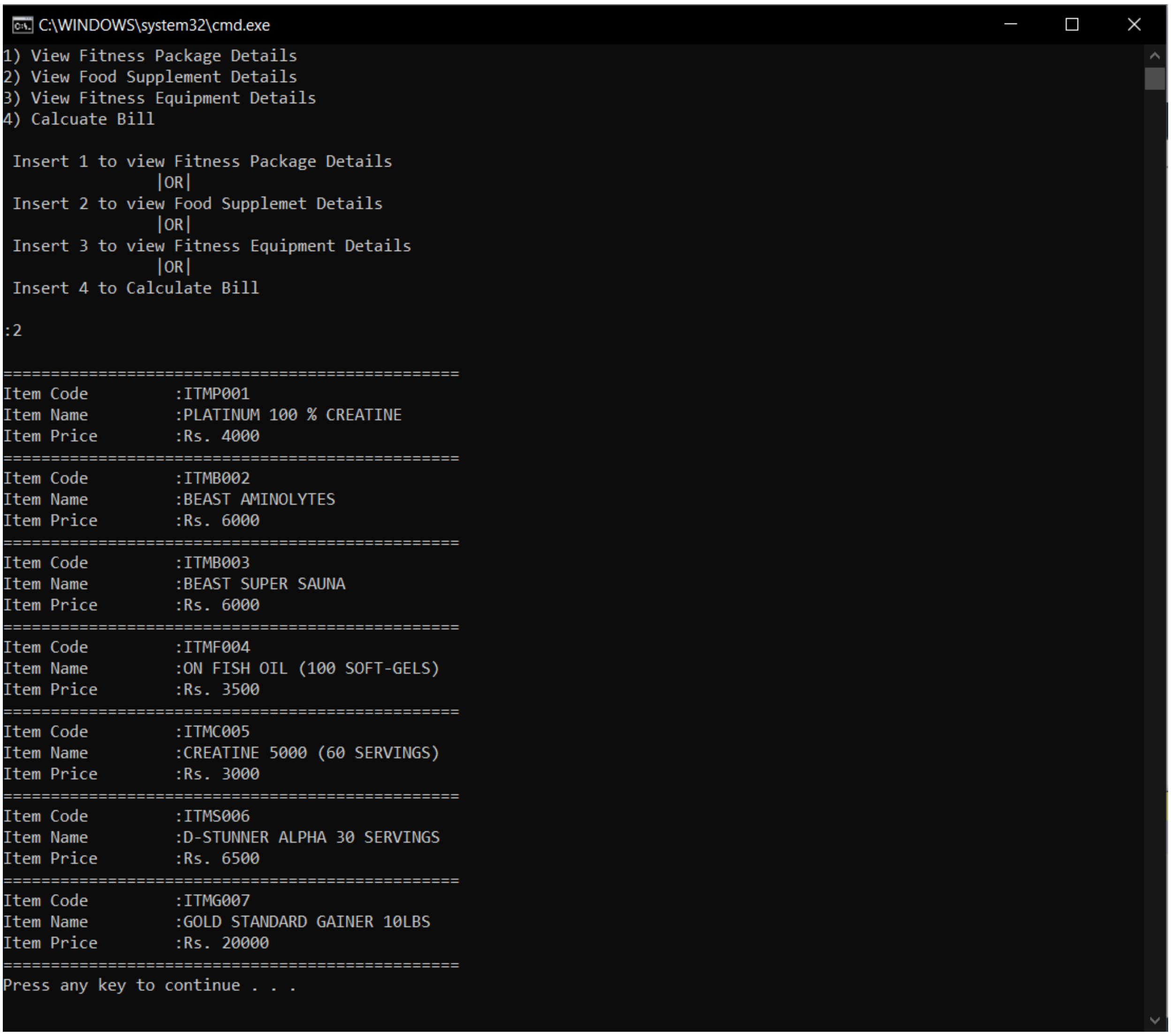
SCREENSHOT	
	
	`Press A for AdminLogin or Press C for CashierLogin : A`
	`Enter Admin's ID : Admin`
	`Enter Admin's Password: 123`
	`****************************************************`
	`WELCOME TO RATHNAYAKA GYM'S MAIN MENU !`
	`****************************************************`
	`1) View Fitness Package Details`
	`2) View Food Supplement Details`
	`3) View Fitness Equipment Details`
	`4) Calcuate Bill`
	`Insert 1 to view Fitness Package Details`
	`|OR|`
	`Insert 2 to view Food Supplemet Details`
	`|OR|`
	`Insert 3 to view Fitness Equipment Details`
	`|OR|`
	`Insert 4 to Calculate Bill`
	`:`

STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC3.1
TEST CASE NAME	Main Menu
SCENARIO	The purpose of conducting this test case is to check whether the view fitness package details function gives the output when 1 is entered
EXPECTED RESULT	Should display all the fitness packages with relevant data assigned for each Package ID, Package Name, Membership Rate, Introductory Rate
ACTUAL RESULT	All the available fitness packages and their details are displayed well
SCREENSHOT	

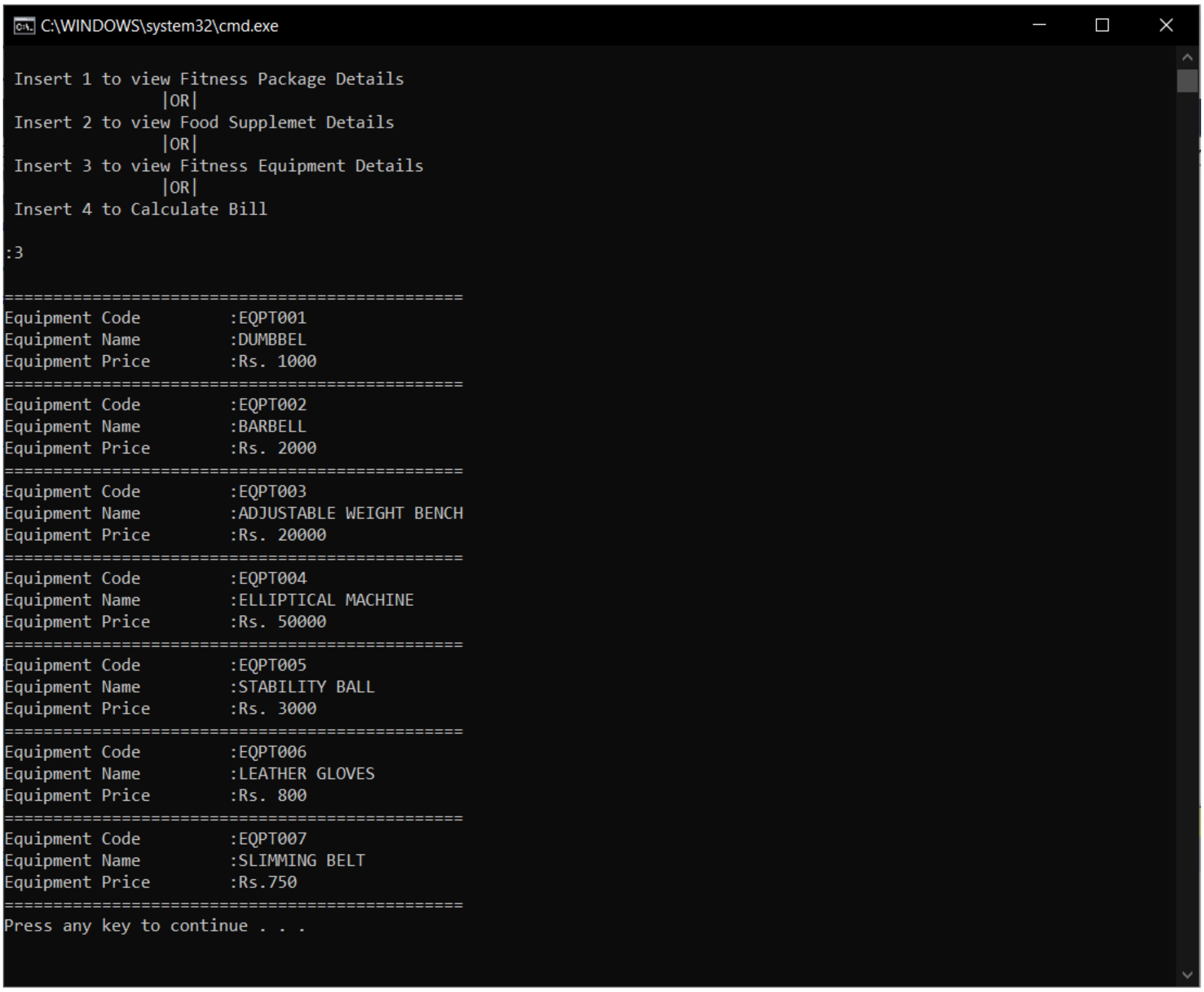
```
1) View Fitness Package Details
2) View Food Supplement Details
3) View Fitness Equipment Details
4) Calcuate Bill

 Insert 1 to view Fitness Package Details
               |OR|
 Insert 2 to view Food Supplemet Details
               |OR|
 Insert 3 to view Fitness Equipment Details
               |OR|
 Insert 4 to Calculate Bill

:1

=====================================================
Package ID       :PKGDT001
Package Name      :DAY WORKOUT
Membership Rate   :Rs. 1500
Introductory Rate :Rs. 1500
=====================================================
Package ID       :PKGDT002
Package Name      :EXECUTIVE MEM (FUL)
Membership Rate   :Rs. 5500
Introductory Rate :Rs. 5500
=====================================================
Package ID       :PKGDT003
Package Name      :1 MONTH MEMBERSHIP
Membership Rate   :Rs. 3000
Introductory Rate :Rs. 3000
=====================================================
Package ID       :PKGDT004
Package Name      :MASTERS MEMBERSHIP OVER 60 YEARS
Membership Rate   :Rs. 5000
Introductory Rate :Rs. 5000
=====================================================
Package ID       :PKGDT005
Package Name      :NON-PEAK EXECUTIVE
Membership Rate   :Rs. 3300
Introductory Rate :Rs. 3300
=====================================================
Press any key to continue . . .
```

STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC3.2
TEST CASE NAME	Main Menu
SCENARIO	The purpose of conducting this test case is to check whether the view food supplement details function gives the output when 2 is entered
EXPECTED RESULT	Should display all the food supplement details with relevant data assigned for each Item Code, Item Name, Item Price
ACTUAL RESULT	All the available food supplements and their details are displayed well
SCREENSHOT	
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

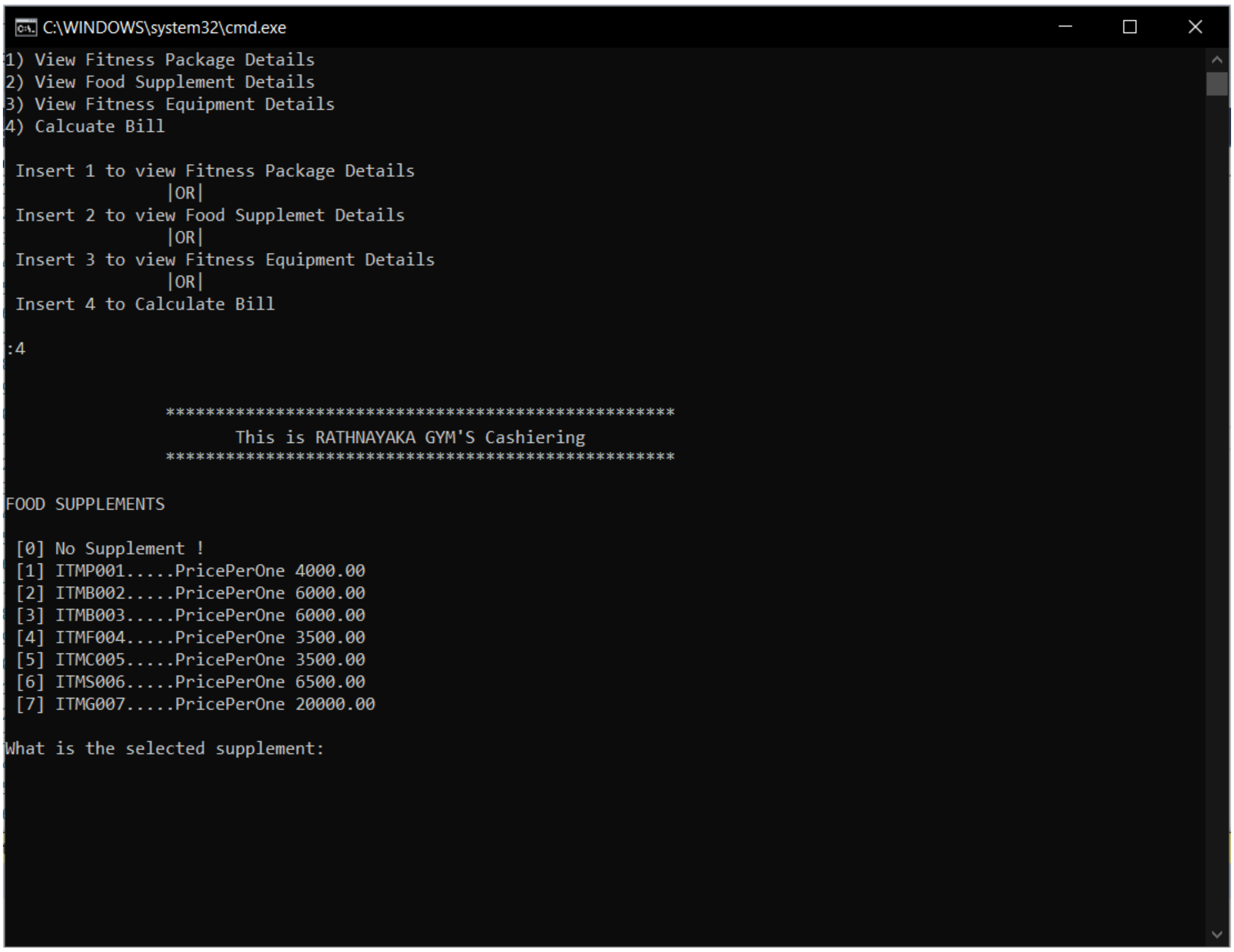
C:\WINDOWS\system32\cmd.exe

```
1) View Fitness Package Details
2) View Food Supplement Details
3) View Fitness Equipment Details
4) Calcuate Bill

 Insert 1 to view Fitness Package Details
                |OR|
 Insert 2 to view Food Supplemet Details
                |OR|
 Insert 3 to view Fitness Equipment Details
                |OR|
 Insert 4 to Calculate Bill

:2

================================================
Item Code         :ITMP001
Item Name         :PLATINUM 100 % CREATINE
Item Price        :Rs. 4000
================================================
Item Code         :ITMB002
Item Name         :BEAST AMINOLYTES
Item Price        :Rs. 6000
================================================
Item Code         :ITMB003
Item Name         :BEAST SUPER SAUNA
Item Price        :Rs. 6000
================================================
Item Code         :ITMF004
Item Name         :ON FISH OIL (100 SOFT-GELS)
Item Price        :Rs. 3500
================================================
Item Code         :ITMC005
Item Name         :CREATINE 5000 (60 SERVINGS)
Item Price        :Rs. 3000
================================================
Item Code         :ITMS006
Item Name         :D-STUNNER ALPHA 30 SERVINGS
Item Price        :Rs. 6500
================================================
Item Code         :ITMG007
Item Name         :GOLD STANDARD GAINER 10LBS
Item Price        :Rs. 20000
================================================
Press any key to continue . . .
```

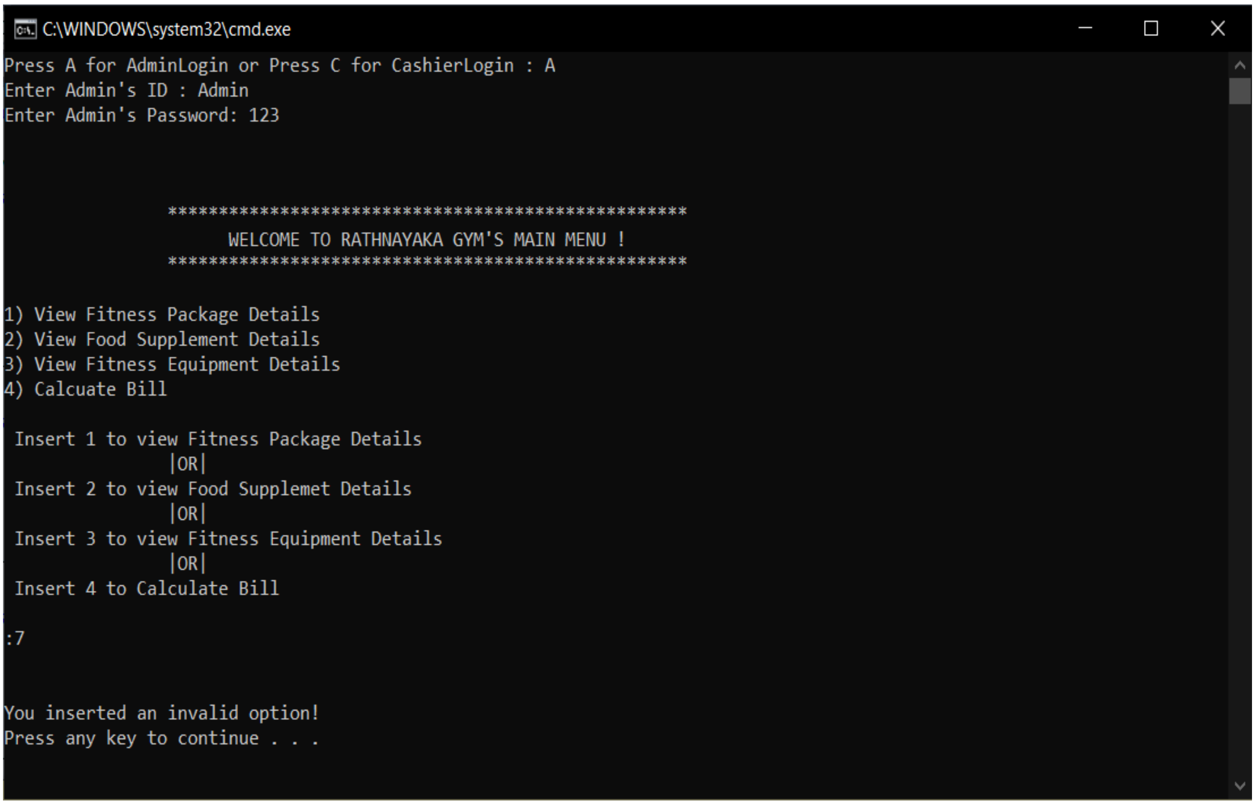
TEST CASE NUMBER	TC3.3
TEST CASE NAME	Main Menu
SCENARIO	The purpose of conducting this test case is to check whether the view fitness equipment details function gives the output when 3 is entered
EXPECTED RESULT	Should display all the fitness equipment details with relevant data assigned for each Equipment Code, Equipment Name, Equipment Price
ACTUAL RESULT	All the available fitness equipment and their details are displayed well
SCREENSHOT	

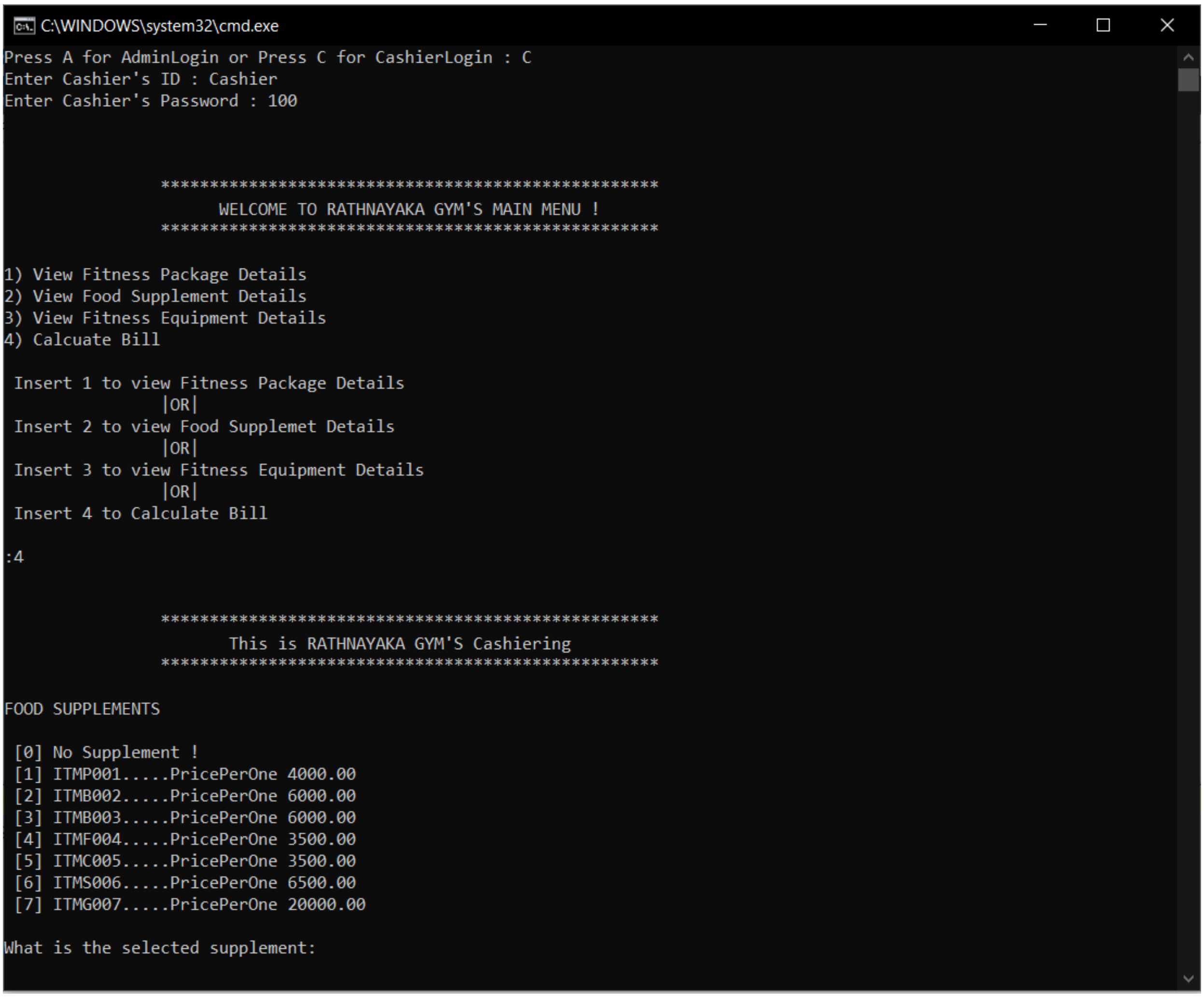
```
 Insert 1 to view Fitness Package Details
               |OR|
 Insert 2 to view Food Supplemet Details
               |OR|
 Insert 3 to view Fitness Equipment Details
               |OR|
 Insert 4 to Calculate Bill

:3


===============================================
Equipment Code          :EQPT001
Equipment Name          :DUMBBEL
Equipment Price         :Rs. 1000
===============================================
Equipment Code          :EQPT002
Equipment Name          :BARBELL
Equipment Price         :Rs. 2000
===============================================
Equipment Code          :EQPT003
Equipment Name          :ADJUSTABLE WEIGHT BENCH
Equipment Price         :Rs. 20000
===============================================
Equipment Code          :EQPT004
Equipment Name          :ELLIPTICAL MACHINE
Equipment Price         :Rs. 50000
===============================================
Equipment Code          :EQPT005
Equipment Name          :STABILITY BALL
Equipment Price         :Rs. 3000
===============================================
Equipment Code          :EQPT006
Equipment Name          :LEATHER GLOVES
Equipment Price         :Rs. 800
===============================================
Equipment Code          :EQPT007
Equipment Name          :SLIMMING BELT
Equipment Price         :Rs.750
===============================================
Press any key to continue . . .
```

STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC3.4
TEST CASE NAME	Main Menu
SCENARIO	The purpose of conducting this test case is to check whether the calculate bill function gives the output when 4 is entered
EXPECTED RESULT	Should display “This is RATHNAYAKA GYM'S Cashiering” and should work the functions one by one which are assigned for calculating bill
ACTUAL RESULT	The statement “This is RATHNAYAKA GYM'S Cashiering” is displayed and the beginning of the bill started
SCREENSHOT	
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

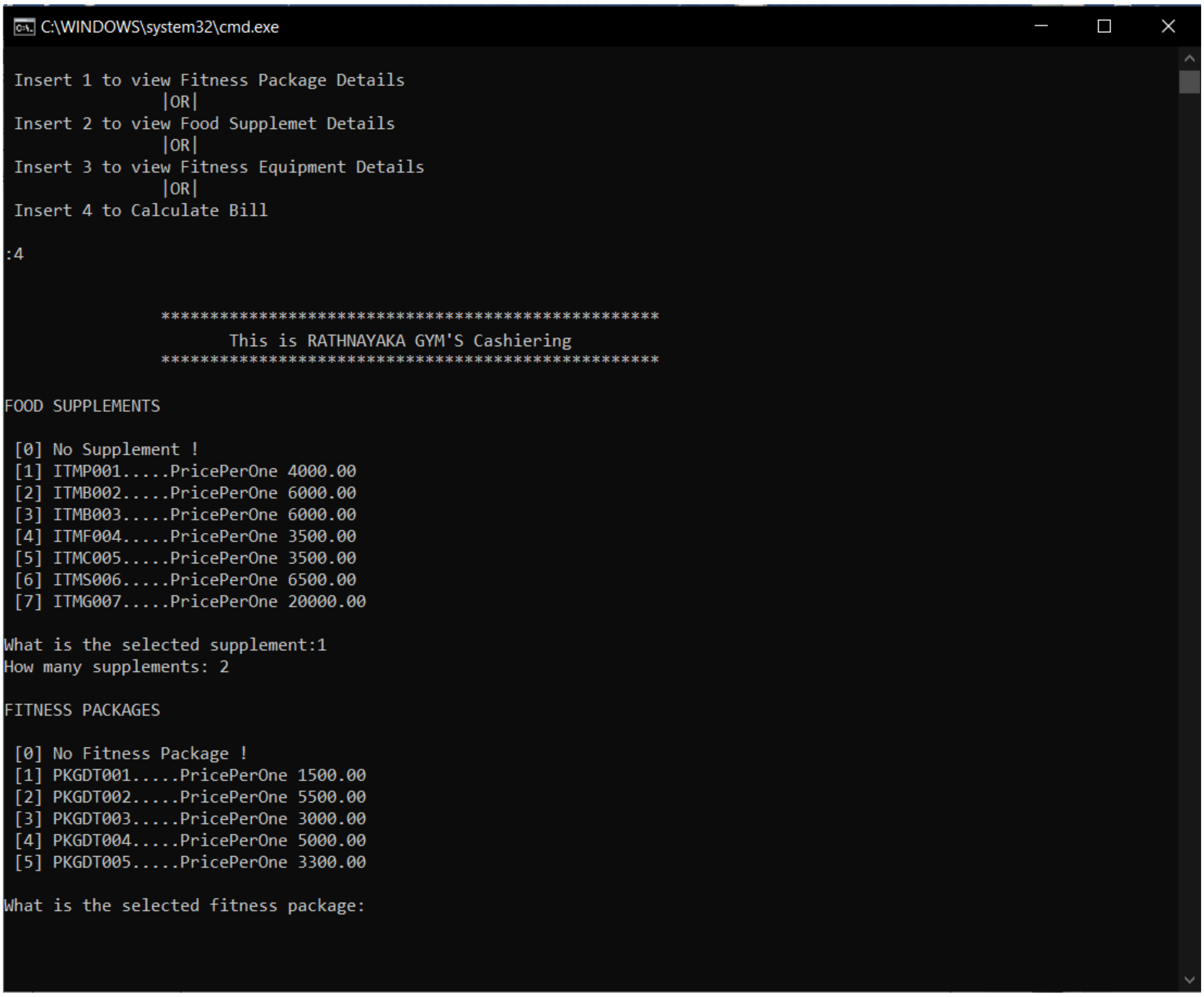
```
1) View Fitness Package Details
2) View Food Supplement Details
3) View Fitness Equipment Details
4) Calcuate Bill

 Insert 1 to view Fitness Package Details
                |OR|
 Insert 2 to view Food Supplemet Details
                |OR|
 Insert 3 to view Fitness Equipment Details
                |OR|
 Insert 4 to Calculate Bill

:4



                ***************************************************
                       This is RATHNAYAKA GYM'S Cashiering
                ***************************************************


FOOD SUPPLEMENTS

 [0] No Supplement !
 [1] ITMP001.....PricePerOne 4000.00
 [2] ITMB002.....PricePerOne 6000.00
 [3] ITMB003.....PricePerOne 6000.00
 [4] ITMF004.....PricePerOne 3500.00
 [5] ITMC005.....PricePerOne 3500.00
 [6] ITMS006.....PricePerOne 6500.00
 [7] ITMG007.....PricePerOne 20000.00

What is the selected supplement:
```

TEST CASE NUMBER	TC3.5
TEST CASE NAME	Main Menu
SCENARIO	The purpose of conducting this test case is to check whether the functions inside the main menu gives the correct outputs even when different number except 1,2,3 & 4 is entered
EXPECTED RESULT	Should display “You inserted an invalid option! Press any key to continue . . .”
ACTUAL RESULT	`You inserted an invalid option! Press any key to continue . . .`
SCREENSHOT	 Press A for AdminLogin or Press C for CashierLogin : A Enter Admin's ID : Admin Enter Admin's Password: 123 ** WELCOME TO RATHNAYAKA GYM'S MAIN MENU ! ** 1) View Fitness Package Details 2) View Food Supplement Details 3) View Fitness Equipment Details 4) Calcuate Bill Insert 1 to view Fitness Package Details |OR| Insert 2 to view Food Supplemet Details |OR| Insert 3 to view Fitness Equipment Details |OR| Insert 4 to Calculate Bill :7 You inserted an invalid option! Press any key to continue . . .
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC4
TEST CASE NAME	Calculate Bill
SCENARIO	The purpose of conducting this test case is to check whether the function of calculating bill works when 4 is entered and display food supplement codes along with the prices
EXPECTED RESULT	Should Display “This is RATHNAYAKA GYM'S Cashiering” and the food supplements with item code and Price
ACTUAL RESULT	The expected result comes
SCREENSHOT	
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

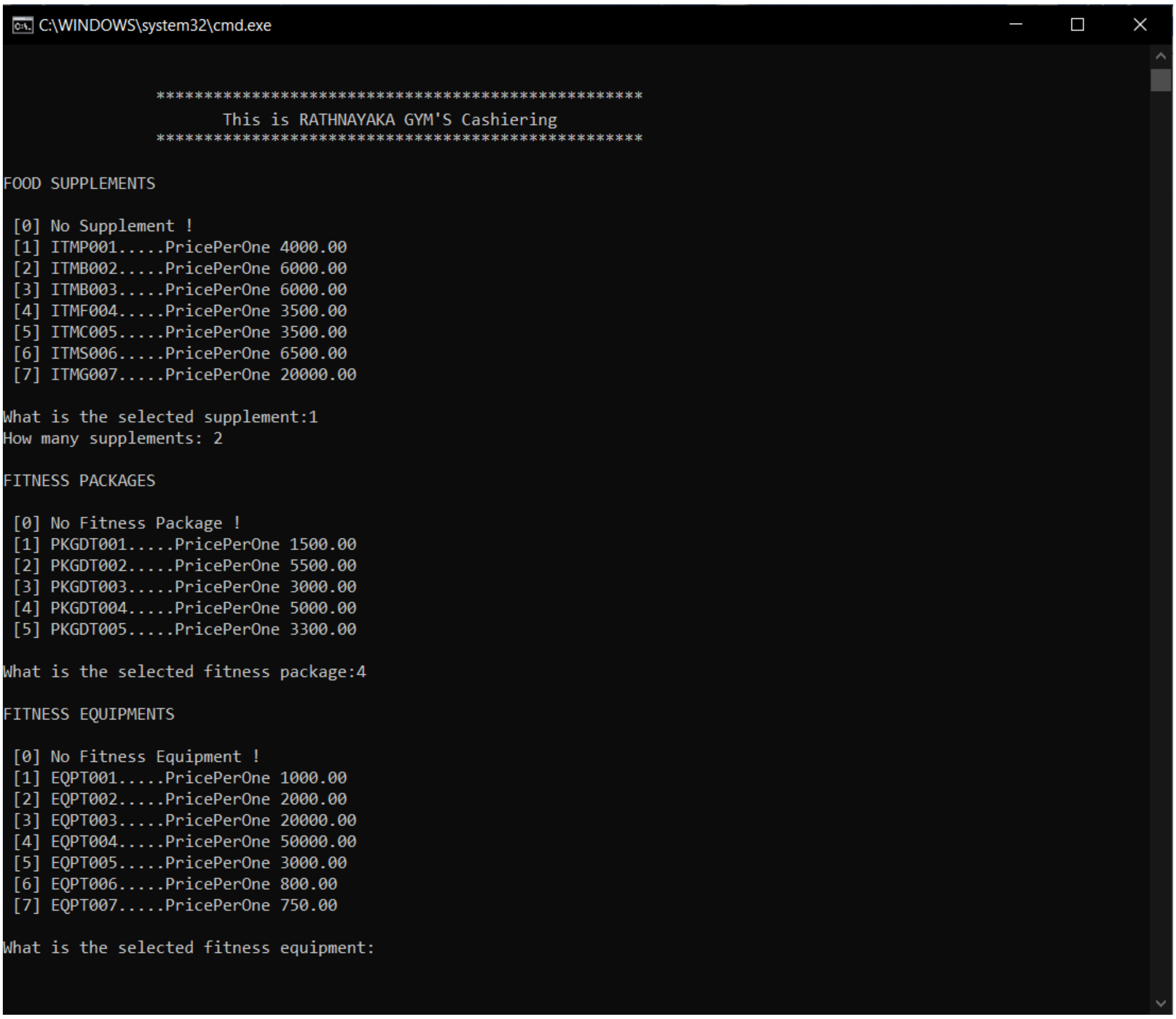
```
Press A for AdminLogin or Press C for CashierLogin : C
Enter Cashier's ID : Cashier
Enter Cashier's Password : 100



                ***************************************************
                      WELCOME TO RATHNAYAKA GYM'S MAIN MENU !
                ***************************************************

1) View Fitness Package Details
2) View Food Supplement Details
3) View Fitness Equipment Details
4) Calcuate Bill

 Insert 1 to view Fitness Package Details
                |OR|
 Insert 2 to view Food Supplemet Details
                |OR|
 Insert 3 to view Fitness Equipment Details
                |OR|
 Insert 4 to Calculate Bill

:4


                ***************************************************
                       This is RATHNAYAKA GYM'S Cashiering
                ***************************************************

FOOD SUPPLEMENTS

 [0] No Supplement !
 [1] ITMP001.....PricePerOne 4000.00
 [2] ITMB002.....PricePerOne 6000.00
 [3] ITMB003.....PricePerOne 6000.00
 [4] ITMF004.....PricePerOne 3500.00
 [5] ITMC005.....PricePerOne 3500.00
 [6] ITMS006.....PricePerOne 6500.00
 [7] ITMG007.....PricePerOne 20000.00

What is the selected supplement:
```

<table>
<tr><td>TEST CASE NUMBER</td><td>TC4.1</td></tr>
<tr><td>TEST CASE NAME</td><td>Calculate Bill (Food Supplements)</td></tr>
<tr><td>SCENARIO</td><td>The purpose of conducting this test case is to check how food supplement details are taken to calculate the bill</td></tr>
<tr><td>EXPECTED RESULT</td><td>Should display the fitness packages with package ID and Price and should display “What is the selected fitness package:” function</td></tr>
<tr><td>ACTUAL RESULT</td><td>The expected result comes</td></tr>
<tr><td>SCREENSHOT</td><td>



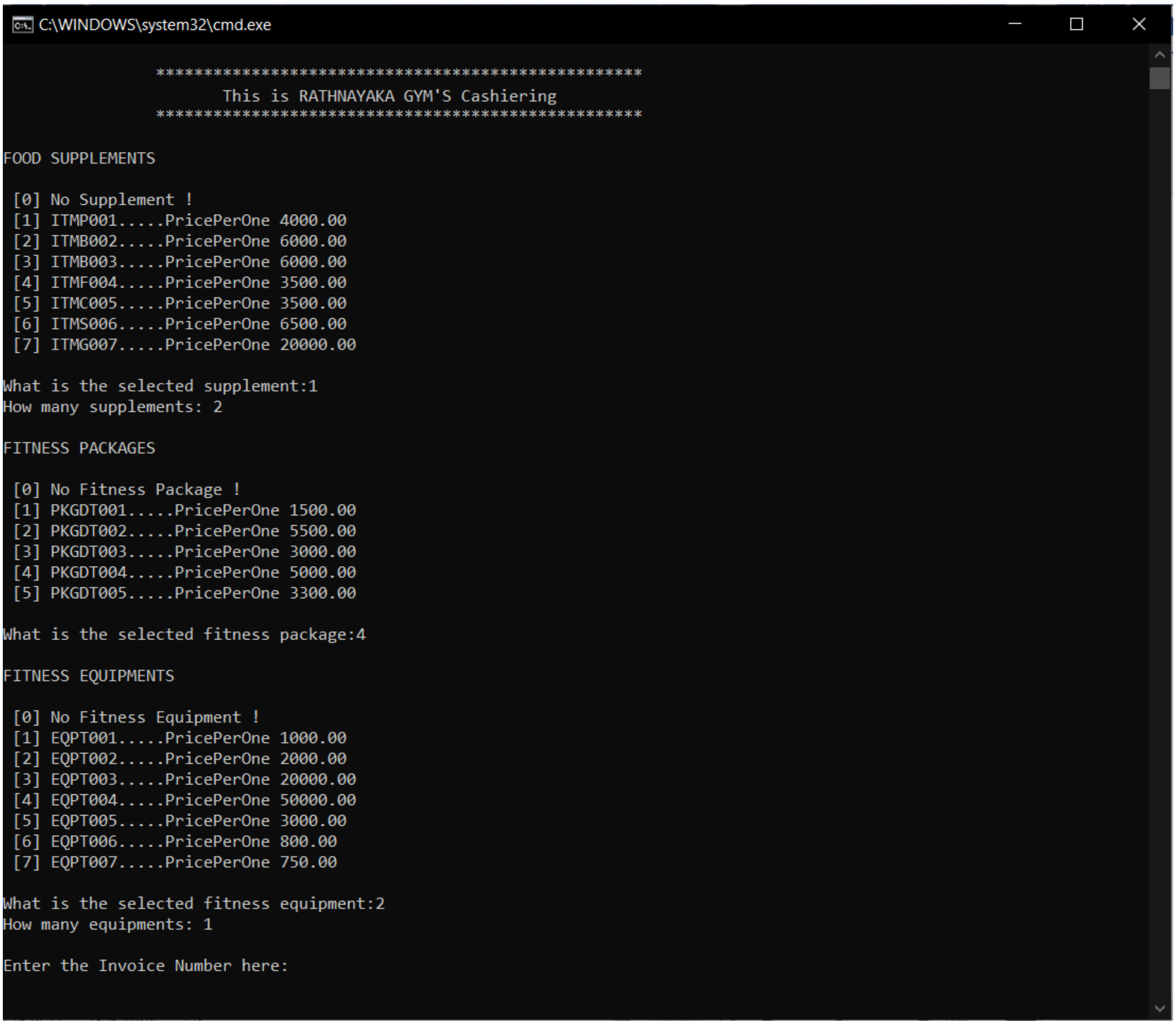
```
 Insert 1 to view Fitness Package Details
                |OR|
 Insert 2 to view Food Supplemet Details
                |OR|
 Insert 3 to view Fitness Equipment Details
                |OR|
 Insert 4 to Calculate Bill

:4



                ***************************************************
                       This is RATHNAYAKA GYM'S Cashiering
                ***************************************************

FOOD SUPPLEMENTS

 [0] No Supplement !
 [1] ITMP001.....PricePerOne 4000.00
 [2] ITMB002.....PricePerOne 6000.00
 [3] ITMB003.....PricePerOne 6000.00
 [4] ITMF004.....PricePerOne 3500.00
 [5] ITMC005.....PricePerOne 3500.00
 [6] ITMS006.....PricePerOne 6500.00
 [7] ITMG007.....PricePerOne 20000.00

What is the selected supplement:1
How many supplements: 2

FITNESS PACKAGES

 [0] No Fitness Package !
 [1] PKGDT001.....PricePerOne 1500.00
 [2] PKGDT002.....PricePerOne 5500.00
 [3] PKGDT003.....PricePerOne 3000.00
 [4] PKGDT004.....PricePerOne 5000.00
 [5] PKGDT005.....PricePerOne 3300.00

What is the selected fitness package:
```

</td></tr>
<tr><td>STATUS (PASS/FAIL)</td><td>Pass</td></tr>
<tr><td>CONCLUSION</td><td>The output comes same as expected</td></tr>
</table>

TEST CASE NUMBER	TC4.2
TEST CASE NAME	Calculate Bill (Fitness Packages)
SCENARIO	The purpose of conducting this test case is to check how fitness package details are taken to calculate the bill
EXPECTED RESULT	Should display the fitness equipment with equipment ID and Price and should display “What is the selected fitness equipment:” function
ACTUAL RESULT	The expected result comes
SCREENSHOT	

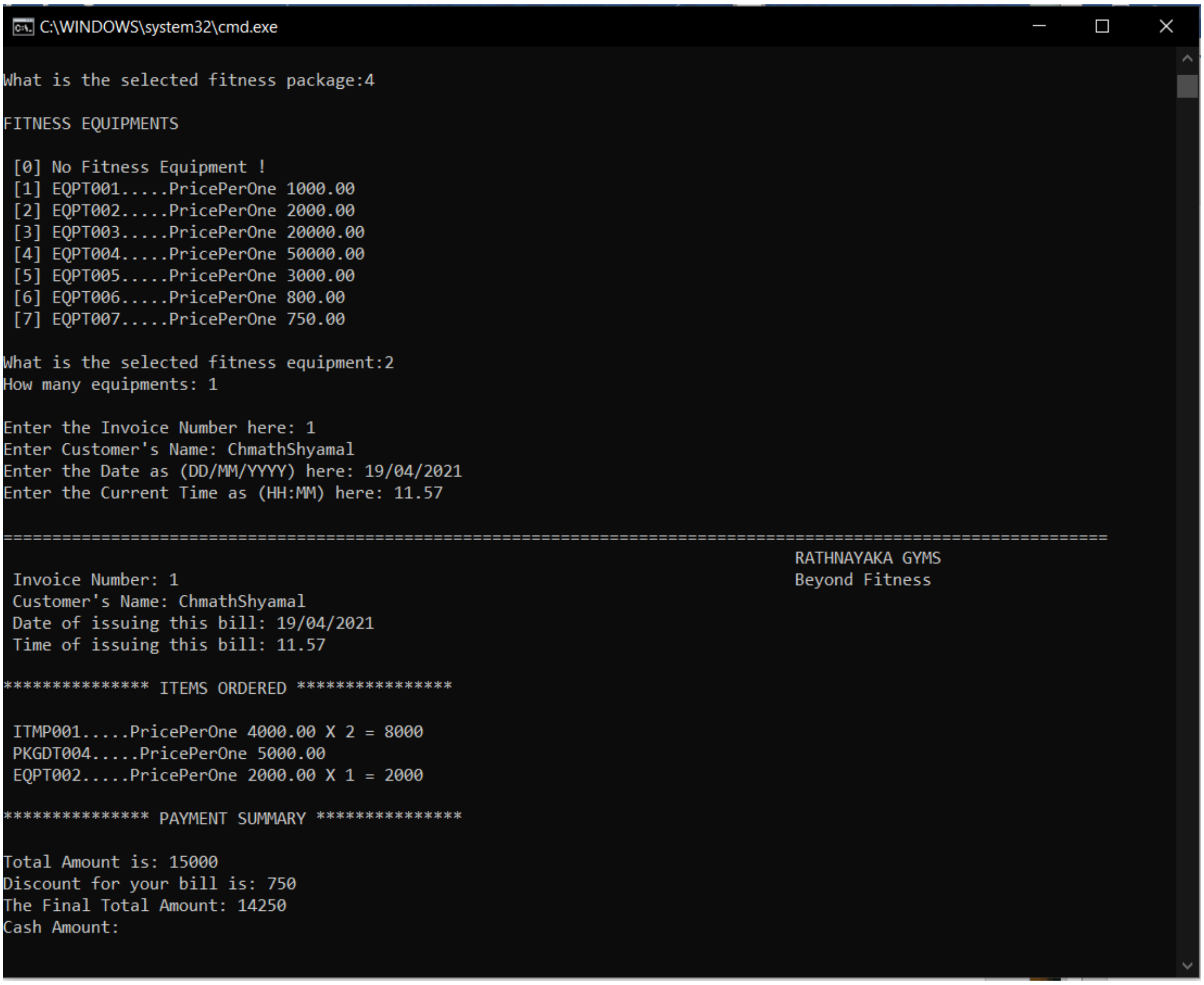
```
                ***************************************************
                       This is RATHNAYAKA GYM'S Cashiering
                ***************************************************

FOOD SUPPLEMENTS

 [0] No Supplement !
 [1] ITMP001.....PricePerOne 4000.00
 [2] ITMB002.....PricePerOne 6000.00
 [3] ITMB003.....PricePerOne 6000.00
 [4] ITMF004.....PricePerOne 3500.00
 [5] ITMC005.....PricePerOne 3500.00
 [6] ITMS006.....PricePerOne 6500.00
 [7] ITMG007.....PricePerOne 20000.00

What is the selected supplement:1
How many supplements: 2

FITNESS PACKAGES

 [0] No Fitness Package !
 [1] PKGDT001.....PricePerOne 1500.00
 [2] PKGDT002.....PricePerOne 5500.00
 [3] PKGDT003.....PricePerOne 3000.00
 [4] PKGDT004.....PricePerOne 5000.00
 [5] PKGDT005.....PricePerOne 3300.00

What is the selected fitness package:4

FITNESS EQUIPMENTS

 [0] No Fitness Equipment !
 [1] EQPT001.....PricePerOne 1000.00
 [2] EQPT002.....PricePerOne 2000.00
 [3] EQPT003.....PricePerOne 20000.00
 [4] EQPT004.....PricePerOne 50000.00
 [5] EQPT005.....PricePerOne 3000.00
 [6] EQPT006.....PricePerOne 800.00
 [7] EQPT007.....PricePerOne 750.00

What is the selected fitness equipment:
```

STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

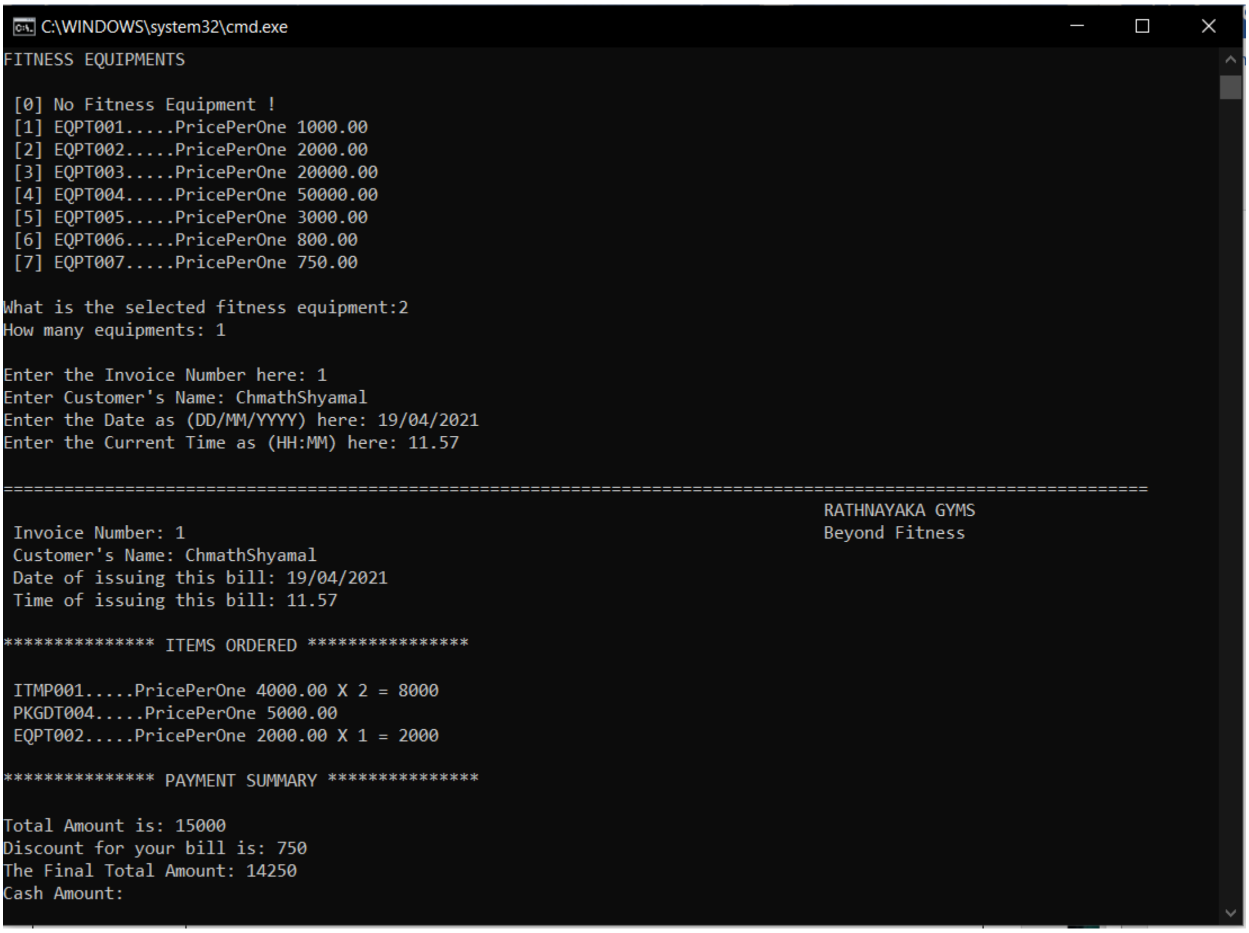
TEST CASE NUMBER	TC4.3
TEST CASE NAME	Calculate Bill (Fitness Equipment)
SCENARIO	The purpose of conducting this test case is to check how fitness equipment details are taken to calculate the bill
EXPECTED RESULT	Should display the “Enter the Invoice Number here:” function
ACTUAL RESULT	The expected result comes
SCREENSHOT	
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

C:\WINDOWS\system32\cmd.exe

```
                ***************************************************
                       This is RATHNAYAKA GYM'S Cashiering
                ***************************************************

FOOD SUPPLEMENTS

 [0] No Supplement !
 [1] ITMP001.....PricePerOne 4000.00
 [2] ITMB002.....PricePerOne 6000.00
 [3] ITMB003.....PricePerOne 6000.00
 [4] ITMF004.....PricePerOne 3500.00
 [5] ITMC005.....PricePerOne 3500.00
 [6] ITMS006.....PricePerOne 6500.00
 [7] ITMG007.....PricePerOne 20000.00

What is the selected supplement:1
How many supplements: 2

FITNESS PACKAGES

 [0] No Fitness Package !
 [1] PKGDT001.....PricePerOne 1500.00
 [2] PKGDT002.....PricePerOne 5500.00
 [3] PKGDT003.....PricePerOne 3000.00
 [4] PKGDT004.....PricePerOne 5000.00
 [5] PKGDT005.....PricePerOne 3300.00

What is the selected fitness package:4

FITNESS EQUIPMENTS

 [0] No Fitness Equipment !
 [1] EQPT001.....PricePerOne 1000.00
 [2] EQPT002.....PricePerOne 2000.00
 [3] EQPT003.....PricePerOne 20000.00
 [4] EQPT004.....PricePerOne 50000.00
 [5] EQPT005.....PricePerOne 3000.00
 [6] EQPT006.....PricePerOne 800.00
 [7] EQPT007.....PricePerOne 750.00

What is the selected fitness equipment:2
How many equipments: 1

Enter the Invoice Number here:
```

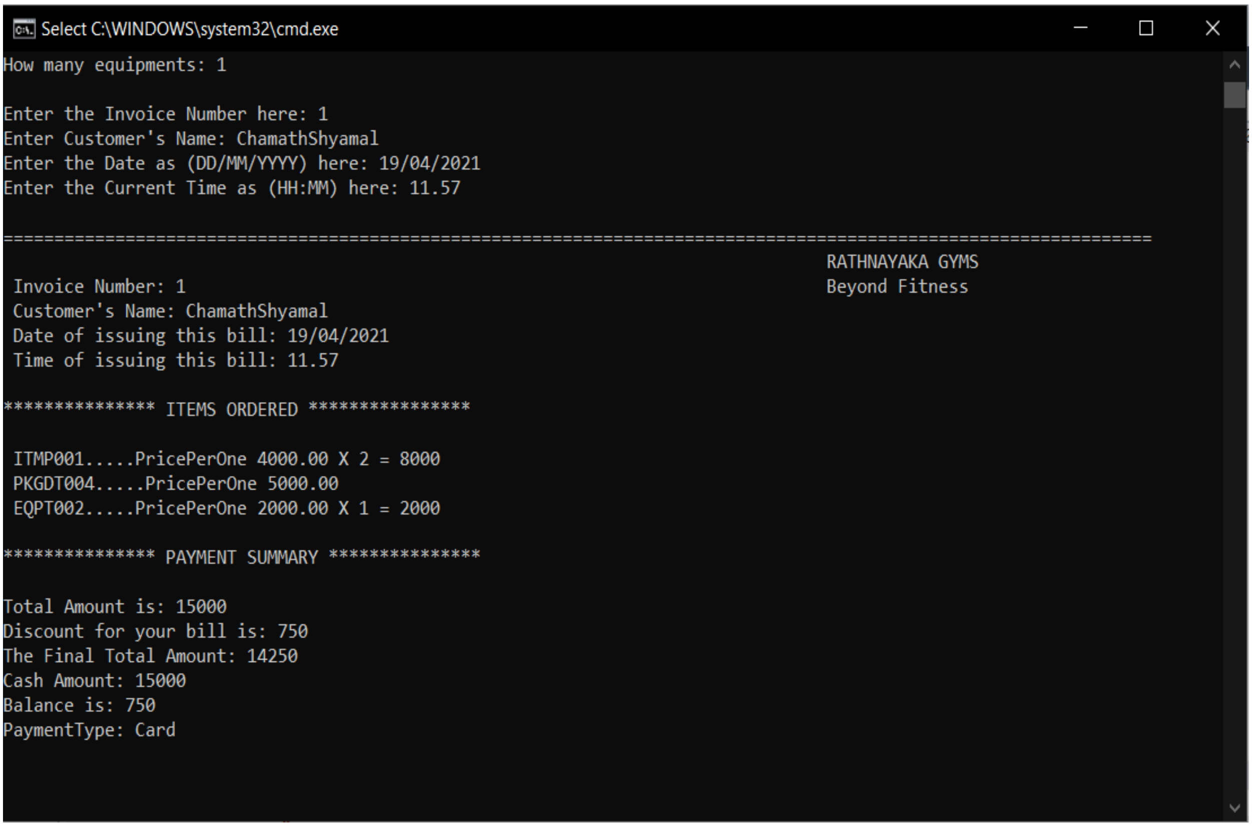
TEST CASE NUMBER	TC4.4
TEST CASE NAME	Calculate Bill
SCENARIO	The purpose of conducting this test case is to check how Invoice number, Customer’s name, Date, Time & Rathnayaka Gym’s slogan is displayed
EXPECTED RESULT	Should display Invoice number, Customer’s name, Date & Time in order with Rathnayaka Gyms slogan at the right bottom & then the Items Ordered
ACTUAL RESULT	The expected result comes
SCREENSHOT	
STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

```
What is the selected fitness package:4

FITNESS EQUIPMENTS

 [0] No Fitness Equipment !
 [1] EQPT001.....PricePerOne 1000.00
 [2] EQPT002.....PricePerOne 2000.00
 [3] EQPT003.....PricePerOne 20000.00
 [4] EQPT004.....PricePerOne 50000.00
 [5] EQPT005.....PricePerOne 3000.00
 [6] EQPT006.....PricePerOne 800.00
 [7] EQPT007.....PricePerOne 750.00

What is the selected fitness equipment:2
How many equipments: 1

Enter the Invoice Number here: 1
Enter Customer's Name: ChmathShyamal
Enter the Date as (DD/MM/YYYY) here: 19/04/2021
Enter the Current Time as (HH:MM) here: 11.57

=================================================================================================================
                                                                                RATHNAYAKA GYMS
 Invoice Number: 1                                                              Beyond Fitness
 Customer's Name: ChmathShyamal
 Date of issuing this bill: 19/04/2021
 Time of issuing this bill: 11.57

*************** ITEMS ORDERED ****************

 ITMP001.....PricePerOne 4000.00 X 2 = 8000
 PKGDT004.....PricePerOne 5000.00
 EQPT002.....PricePerOne 2000.00 X 1 = 2000

*************** PAYMENT SUMMARY ***************

Total Amount is: 15000
Discount for your bill is: 750
The Final Total Amount: 14250
Cash Amount:
```

TEST CASE NUMBER	TC4.5
TEST CASE NAME	Calculate Bill (Payment Summary)
SCENARIO	The purpose of conducting this test case is to check how the system display Total amount, Discount, the final total & the amount to be paid
EXPECTED RESULT	Should display Total Amount, Discount, The Final Total Amount after deducting the discount from the Total & Cash Amount that the customer going to pay
ACTUAL RESULT	The expected result comes
SCREENSHOT	

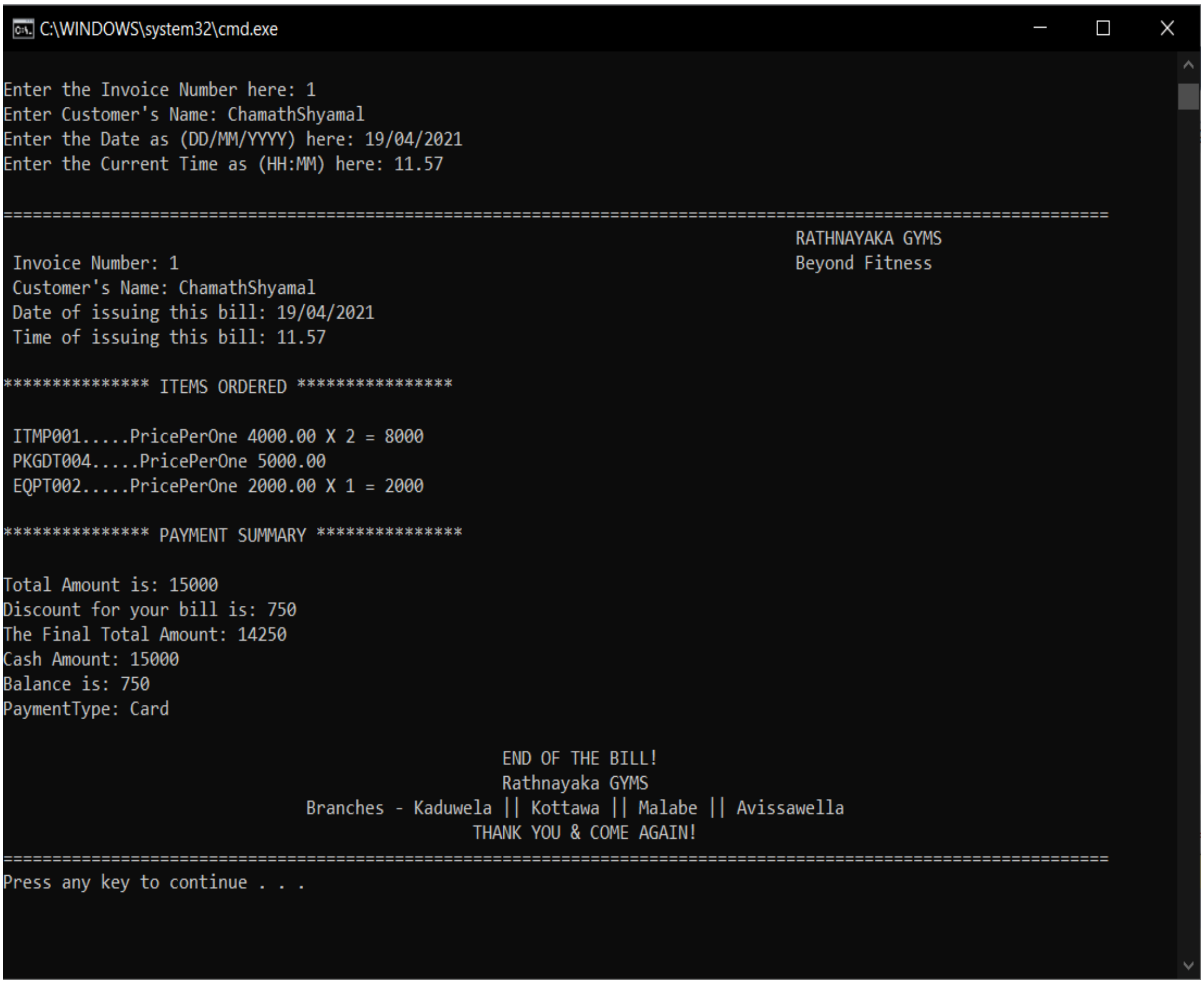
```
FITNESS EQUIPMENTS

 [0] No Fitness Equipment !
 [1] EQPT001.....PricePerOne 1000.00
 [2] EQPT002.....PricePerOne 2000.00
 [3] EQPT003.....PricePerOne 20000.00
 [4] EQPT004.....PricePerOne 50000.00
 [5] EQPT005.....PricePerOne 3000.00
 [6] EQPT006.....PricePerOne 800.00
 [7] EQPT007.....PricePerOne 750.00

What is the selected fitness equipment:2
How many equipments: 1

Enter the Invoice Number here: 1
Enter Customer's Name: ChmathShyamal
Enter the Date as (DD/MM/YYYY) here: 19/04/2021
Enter the Current Time as (HH:MM) here: 11.57

==================================================================================================================
                                                                                RATHNAYAKA GYMS
 Invoice Number: 1                                                              Beyond Fitness
 Customer's Name: ChmathShyamal
 Date of issuing this bill: 19/04/2021
 Time of issuing this bill: 11.57

*************** ITEMS ORDERED ****************

 ITMP001.....PricePerOne 4000.00 X 2 = 8000
 PKGDT004.....PricePerOne 5000.00
 EQPT002.....PricePerOne 2000.00 X 1 = 2000

*************** PAYMENT SUMMARY ***************

Total Amount is: 15000
Discount for your bill is: 750
The Final Total Amount: 14250
Cash Amount:
```

STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC4.5.1
TEST CASE NAME	Calculate Bill (Payment Summary)
SCENARIO	The purpose of conducting this test case is to check entering of amount paid by the customer, the balance amount that should return and the payment type
EXPECTED RESULT	Should display Balance amount that should return to customer & should display function to input payment method
ACTUAL RESULT	The expected result comes
SCREENSHOT	

```
How many equipments: 1

Enter the Invoice Number here: 1
Enter Customer's Name: ChamathShyamal
Enter the Date as (DD/MM/YYYY) here: 19/04/2021
Enter the Current Time as (HH:MM) here: 11.57

=================================================================================================================
                                                                                 RATHNAYAKA GYMS
 Invoice Number: 1                                                               Beyond Fitness
 Customer's Name: ChamathShyamal
 Date of issuing this bill: 19/04/2021
 Time of issuing this bill: 11.57

*************** ITEMS ORDERED ****************

 ITMP001.....PricePerOne 4000.00 X 2 = 8000
 PKGDT004.....PricePerOne 5000.00
 EQPT002.....PricePerOne 2000.00 X 1 = 2000

*************** PAYMENT SUMMARY ***************

Total Amount is: 15000
Discount for your bill is: 750
The Final Total Amount: 14250
Cash Amount: 15000
Balance is: 750
PaymentType: Card
```

STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

TEST CASE NUMBER	TC4.5.2
TEST CASE NAME	Calculate Bill (Payment Summary)
SCENARIO	The purpose of conducting this test case is to check the ending of the bill should display “END OF THE BILL! Gym Name (Rathnayaka Gyms), branches and THANK YOU & COME AGAIN! Press any key to continue . . .”
EXPECTED RESULT	Should display “END OF THE BILL!”, Gym Name, branches and “THANK YOU & COME AGAIN! “Press any key to continue . . .
ACTUAL RESULT	The expected result comes
SCREENSHOT	

```
Enter the Invoice Number here: 1
Enter Customer's Name: ChamathShyamal
Enter the Date as (DD/MM/YYYY) here: 19/04/2021
Enter the Current Time as (HH:MM) here: 11.57

==================================================================================================================
                                                                                 RATHNAYAKA GYMS
 Invoice Number: 1                                                               Beyond Fitness
 Customer's Name: ChamathShyamal
 Date of issuing this bill: 19/04/2021
 Time of issuing this bill: 11.57

*************** ITEMS ORDERED ****************

 ITMP001.....PricePerOne 4000.00 X 2 = 8000
 PKGDT004.....PricePerOne 5000.00
 EQPT002.....PricePerOne 2000.00 X 1 = 2000

*************** PAYMENT SUMMARY ***************

Total Amount is: 15000
Discount for your bill is: 750
The Final Total Amount: 14250
Cash Amount: 15000
Balance is: 750
PaymentType: Card

                                                   END OF THE BILL!
                                                   Rathnayaka GYMS
                               Branches - Kaduwela || Kottawa || Malabe || Avissawella
                                                THANK YOU & COME AGAIN!
==================================================================================================================
Press any key to continue . . .
```

STATUS (PASS/FAIL)	Pass
CONCLUSION	The output comes same as expected

Conclusion

However, I could get many advantages from this course work such as how to solve doubts and issues, how to gather requirements needed (specially from internet), how to do the analyzation properly, how to manage time effectively, how to overcome from errors which are arising when writing the code and so on. Finally…with all above mentioned details and my proper time management, I could prepare a successful document to complete this assignment. Thus, I hope this assignment been a great help for me to get learnt about Fundamentals in Programming and to prove that I have successfully completed my fifth assignment in my HD Program. Moreover, during this fundamental in programming assessment, I learnt about the basics of programming while learning C++ programming language up to some extent, requirement gathering, system requirement document preparing and many more things. This project gave me the chance to try my new skills in practice. Topics like structured programming, object-oriented programming, sequences, selection and loops were discussed in this document. Moreover, flowcharts, pseudo codes, a test plan, test cases were created ad included in this document. Further, the coding part which include functions of viewing fitness packages, food supplements, fitness equipment details and calculation of bill. So, I have built the application in a user-friendly manner including login part, main menu, option for selections, etc. While doing this project I also gained deep understanding on programming/coding and how it can be implemented in real life situations as now I have the experience in creating an application for Rathnayaka Gyms. Thus, I believe that this was a great chance for me to improve my computer programming skills.

References

Anon., 2021. *Tutorialspoint.* [Online]
Available at:
https://www.tutorialspoint.com/programming_methodologies/programming_methodologies_introduction.htm
[Accessed 5 March 2021].
Dale Janssen, C. J., 2008. *Techopedia.* [Online]
Available at: https://www.techopedia.com/definition-sop
[Accessed 10 March 2021].
Matt Wilwy, J. F., 2020. *Springer Link.* [Online]
Available at:
https://books.google.com/books/about/Introduction_to_Object_Oriented_Programm.html?id=ily2cfpKy6kC
[Accessed 8 March 2021].
Nygaard, K., 1986. *Books.Google-An introduction to object oriented programming.* [Online]
Available at:
https://books.google.com/books/about/Introduction_to_Object_Oriented_Programm.html?id=ily2cfpKy6kC
[Accessed 6 March 2021].
Roy, P. V., 2004. *Books.google.* [Online]
Available at:
https://books.google.com/books/about/Concepts_Techniques_and_Models_of_Comput.html?id=gL34DwAAQBAJ
[Accessed 15 March 2021].
For Basics in Computer Programmng- https://www.guru99.com/computer-programming-tutorial.html
Got few coding helps from - https://www.geeksforgeeks.org/structure-sorting-in-c/ , https://www.studytonight.com/cpp-programs