Semester: 4^{th}

Name: AMIR ABBAS

ID: 15499

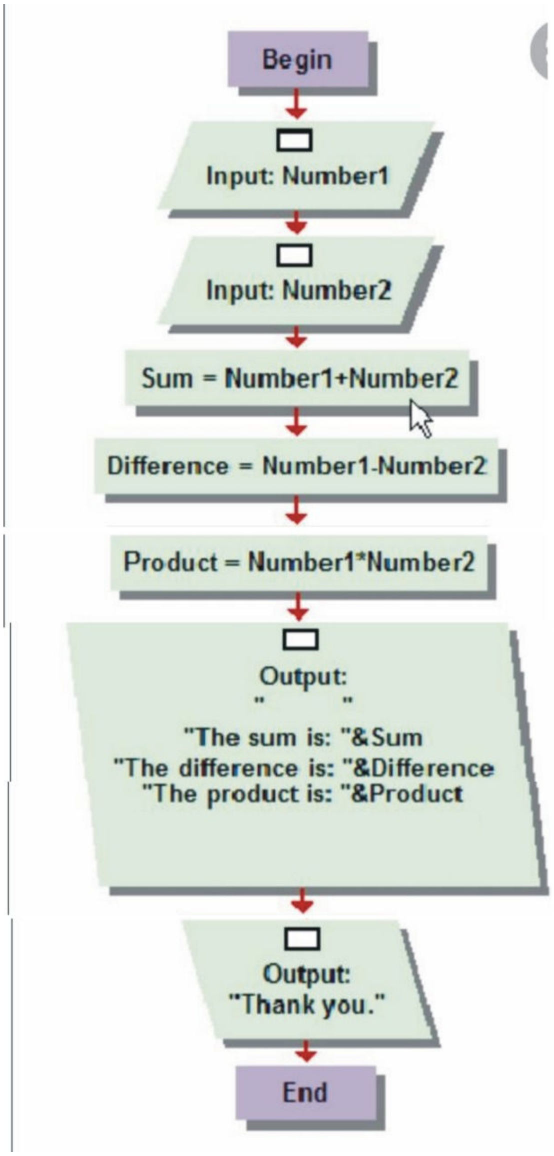
Paper: Programing Fundamentals

Teacher Name: Dr. Fazal E Malik

Q.1 (a)

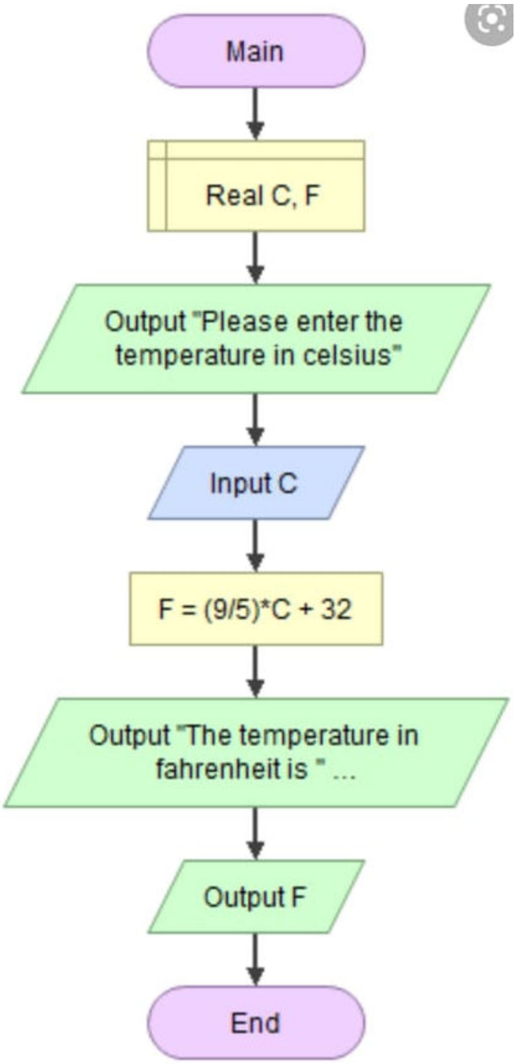
Ans.1 (a)

FLOW CHART:


Begin
Input: Number1
Input: Number2
Sum = Number1+Number2
Difference = Number1-Number2
Product = Number1*Number2
Output:
" "
"The sum is: "&Sum
"The difference is: "&Difference
"The product is: "&Product
Output:
"Thank you."
End

Q.1 (b)

FLOW CHART:


Main
Real C, F
Output "Please enter the temperature in celsius"
Input C
F = (9/5)*C + 32
Output "The temperature in fahrenheit is " ...
Output F
End

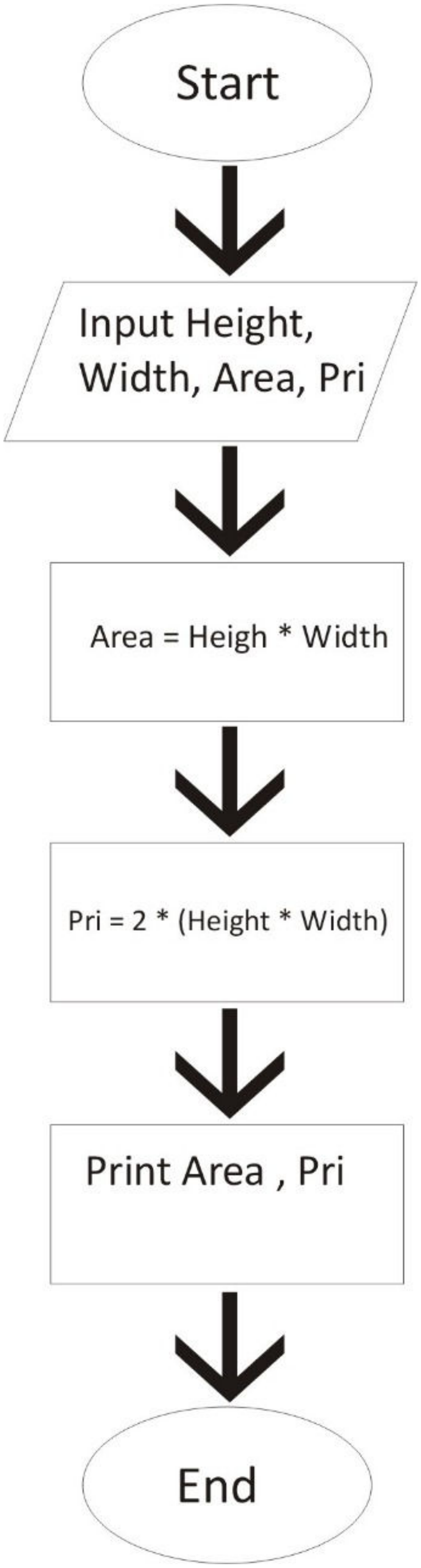
Program Coding:

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

int main()
{
  float fahr, cel;
  char option;
  cout << "Choose from following option:" << endl;
  cout << "1. Celsius to Fahrenheit." << endl;
  cout << "2. Fahrenheit to Celsius." << endl;
  cin >> option;

  //option for converting celsius into fahernheit
  if (option == '1')
  {
    cout << "Enter the temperature in Celsius: ";
    cin >> cel;

    fahr = (1.8 * cel) + 32.0; //temperature conversion formula
    cout << "\nTemperature in degree Fahrenheit: " << fahr << " F" << endl;
  }
  //option for converting Fahrenheit into Celsius
  else if (option == '2')
  {
    cout << "Enter the temperature in Fahrenheit: ";
    cin >> fahr;

    cel = (fahr - 32) / 1.8; //temperature conversion formula
    cout << "\nTemperature in degree Celsius: " << cel << " C" << endl;
  }
  else
    cout << "Error Wrong Input." << endl;

  return 0;
}
```

Q.2 (a)

Ans.2 (a)

Flow Chart:

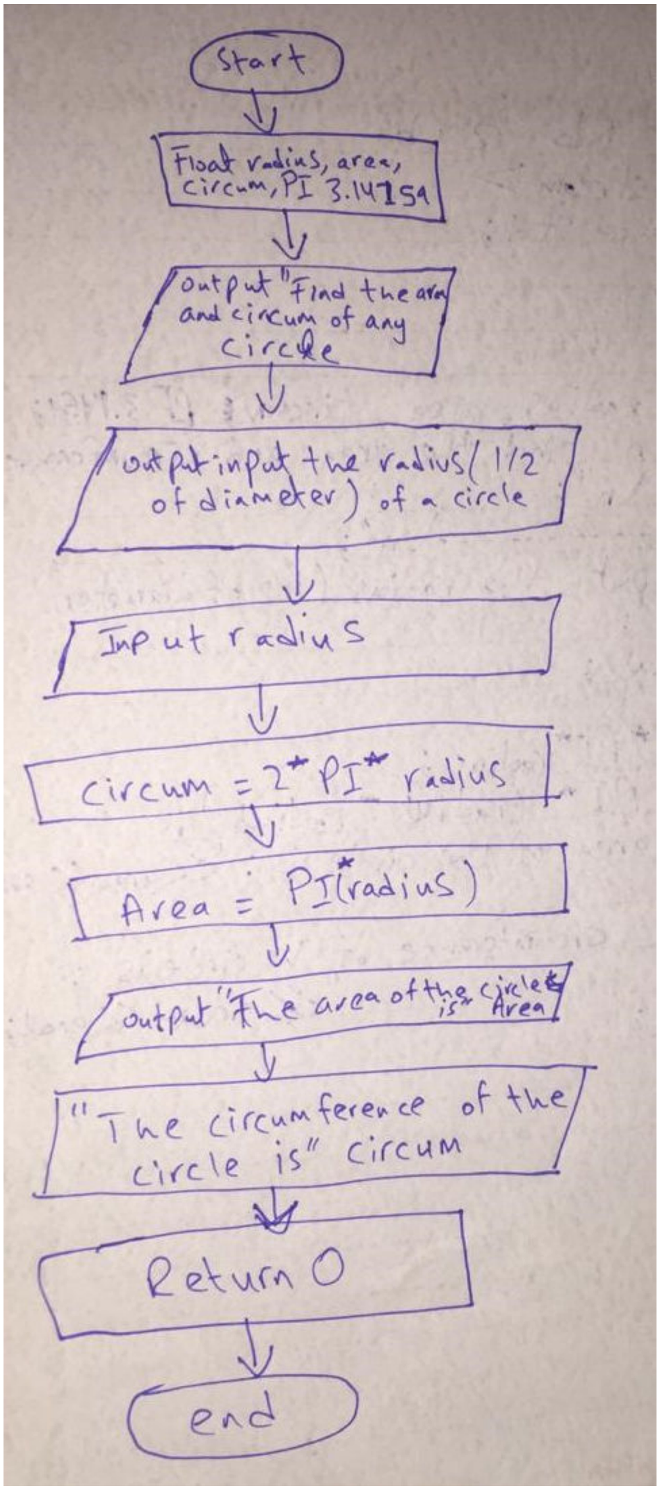


Program Coding:

```
#include<iostream.h>
#include<conio.h>

void main()
{
 clrscr();
 int len, bre, peri, area;
 cout<<"Enter Length and Breadth of the Rectangle: ";
 cin>>len>>bre;
 area=len*bre;
 peri=(2*len)+(2*bre);
 cout<<"Area of Rectangle: "<<area<<"\tPerimeter Rectangle: "<<peri;
 getch();
}
```

Q.2 (b)

Flow Chart:



Program Coding:

```
#include<iostream.h>

#define PI 3.14159

Using namespace std;

Int main()

{

        Float radius, area, circum;

        Cout << “\n\n Find the area and circumference of any circle :\n”;

        Cout << “--------------------------------------------------------------------\n”;

        Cout << “ Input the radius (1/2 of diameter) of a circle :” ;

        Cin>>radius;


        Circum = 2*PI*Radius;

        Area = PI*(radius*radius);

        Cout<< “ The area of the circle is :”<< area << endl;

        Cout<< “ The circumference of the circle is : “<< circum << endl;


        Cout << endl;

        Return 0;

}
```

Q.3 (a)

Programming Languages:

- Programming languages specially developed so that you could pass your data and instructions to the computer to do specific job
- There are two major types of programming languages,
    - Low Level Languages
    - High Level Languages

1. Low Level languages are further divided in to Machine language and Assembly language
2. High Level Languages are, for scientific application FORTRAN and C languages are used. On the other hand COBOL is used for business applications.

Machine Language:

- Machine Language is the only language that is directly understood by the computer. It does not need any translator program
- The only advantage is that program of machine language run very fast
- There is nothing “below” machine language – only hardware.
- Impossible for humans to read. Consists of only 0’s and 1’s.
    - 0001001111110000
- In the earliest days of computers, the only programming languages available were machine languages. Each computer had its own machine language, which was made of streams of 0s and 1s.

Assembly Language:

- The next evolution in programming came with the idea of replacing binary code for instruction and addresses with symbols. Because they used symbols, these languages were first known as symbolic languages. The set of these mnemonic languages were later referred to as assembly languages.

- It is the first step to improve the programming structure, you should know that computer can handle numbers and letter.
- The set of symbols and letters forms the Assembly Language and a translator program is required to translate the Assembly Language to machine language
- This translator program used for Assembly Language is called Assembler
- To program in assembly you need to understand concepts behind machine language and execution-fetch cycle of CPU.
- Assembly is a machine specific language.
- Although Assembly and machine language might look similar, they are in fact two different types of languages.
    - Assembly consists of both binary and simple words
    - Machine code composed only of 0's and 1's

High Level Language:

- Although assembly languages greatly improved programming efficiency, they still required programmers to concentrate on the hardware they were using. Working with symbolic languages was also very tedious, because each machine instruction had to be individually coded. The desire to improve programmer efficiency and to change the focus from the computer to the problem being solved led to the development of high- level languages.
- Assembly and machine level languages require deep knowledge of computer hardware where as in higher language you have to know only the instructions in English words and logic of the problem.
- Higher level languages are simple languages that use English and mathematical symbols like +, -, %, / etc. for its program construction
- Any higher level language has to be converted to machine language for the computer to understand

- For example COBOL (Common Business Oriented Language), FORTRAN (Formula Translation) and BASIC (Beginners All-purpose Symbolic Instruction Code) are high level languages

- Higher level languages have a major advantage over machine and assembly languages that higher level languages are easy to learn and use (similar to the languages used by us in our day to day life.

Q.3 (b)

There are Two translator uses to translate high level language to machine language.

1. **Compiler:**

- It is a program translator that translates the instruction of a higher level language to machine language.

- It is called compiler because it compiles machine language instructions for every program instructions of higher level language.

- Thus compiler is a program translator like assembler but more sophisticated. It scans the entire program first and then translates it into machine code.

- The programs written by the programmer in higher level language is called source program. After this program is converted to machine languages by the compiler it is called object program

- A compiler can translate only those source programs, which have been written, in that language

2. **Interpreter:**

- An interpreter is another type of program translator used for translating higher level language into machine language.

- It takes one statement of higher level languages, translate it into machine language and immediately execute it.

- Translation and execution are carried out for each statement.

- It differs from compiler, which translate the entire source program into machine code.

- The advantage of interpreter compared to compiler is its fast response to changes in source program do not require large memory in computer.
- The disadvantage of interpreter is that it is time consuming method because each time a statement in a program is executed then it is first translated.
- Thus compiled machine language program runs much faster than an interpreted program.