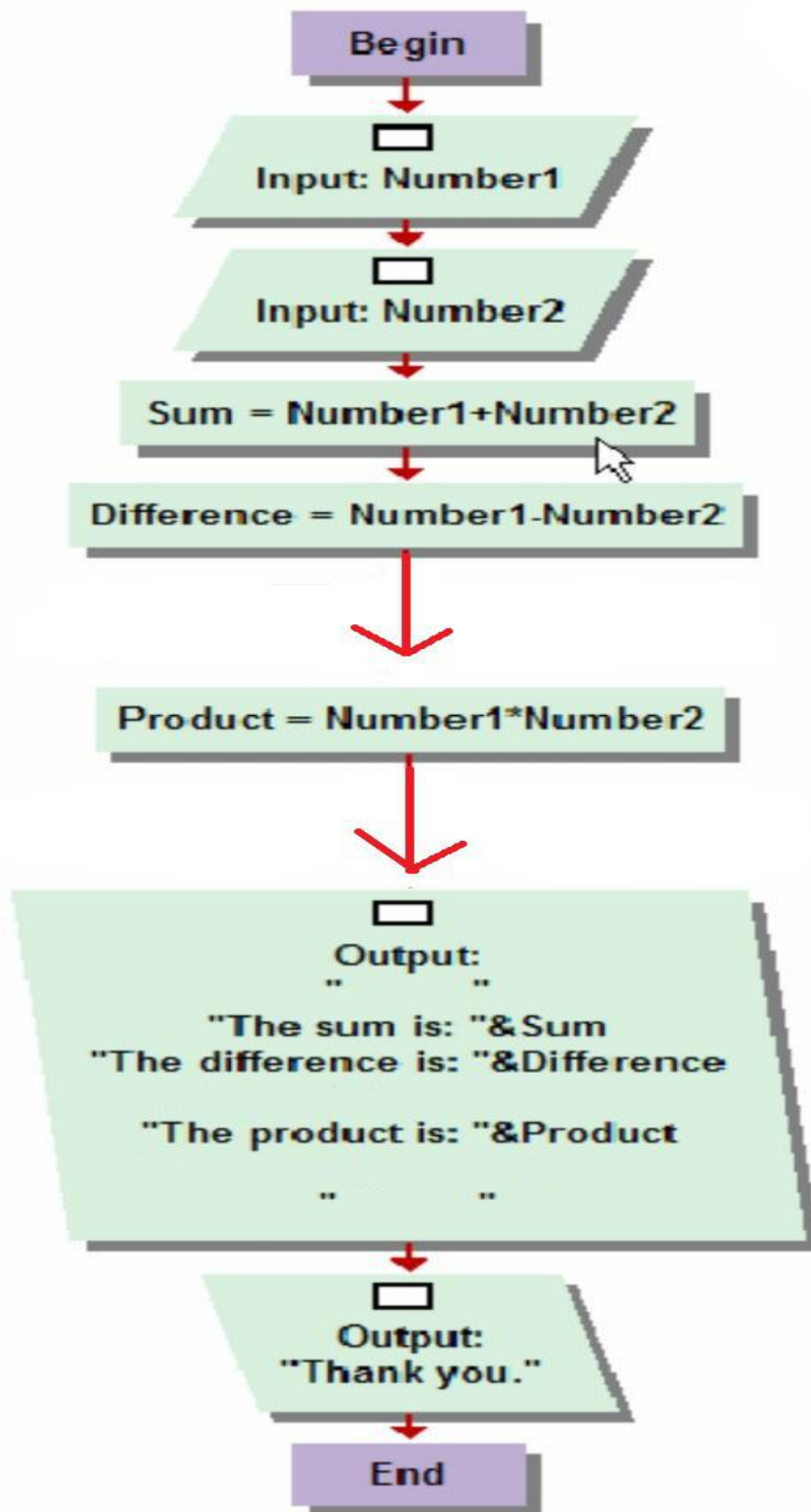
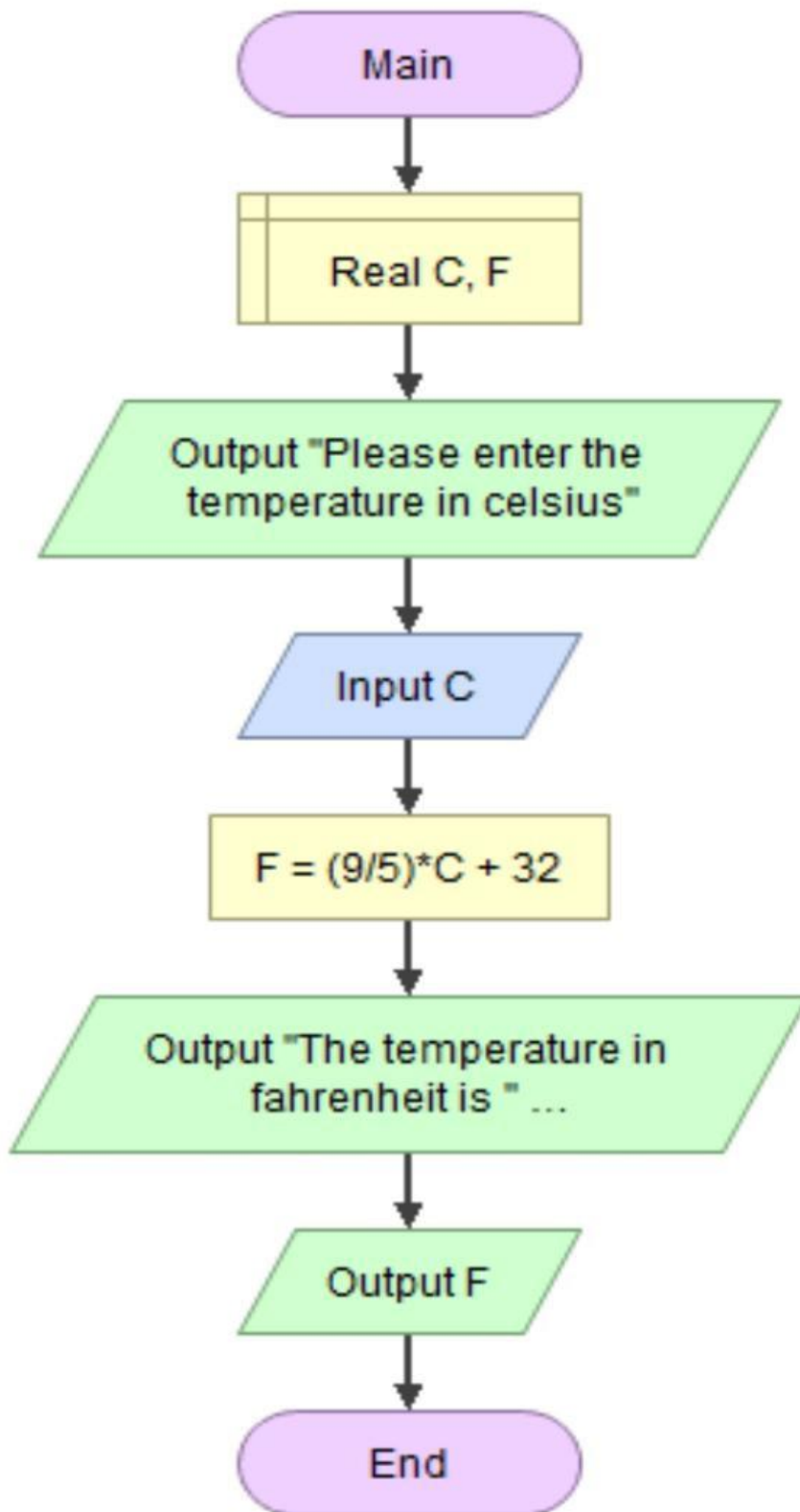


Name: Aftab Khan ID:12985
Subject: Programming Fundamental

Answer 1 (a):



Answer 1 (b):



Answer 2 (a):

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

int main()
{
    int width, length, area, peri;

    cout << "\n\n Find the Area and Perimeter of a Rectangle :\n";

    cout << "-----\n";

    cout<<" Input the length of the rectangle : ";
    cin>>length;

    cout<<" Input the width of the rectangle : ";
    cin>>width;

    area=(length*width);
    peri=2*(length+width);

    cout<<" The area of the rectangle is : "<< area << endl;

    cout<<" The perimeter of the rectangle is : "<< peri << endl;

    cout << endl;

    return 0;

}
```

Answer 2 (b):

Formula: $\text{Area} = 3.14 * \text{radius}^2$

$\text{Circumference} = 2 * 3.14 * \text{radius}$

Program to calculate Area and Circumference based on user input:

To calculate area and circumference we must know the radius of circle. The program will prompt user to enter the radius and based on the input it would calculate the values. To make it simpler we have taken standard PI value as 3.14 (constant) in the program. Other details are mentioned as comments in the below example program.

Code:

```
#include <iostream>
#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;
}
```

Answer 3 (a):

There are three main kinds of programming language:

1) Machine language:

Sometimes machine language referred to as machine code or object code, machine language is a collection of binary digits or bits that the computer reads and interprets. Machine language is the only language a computer is capable of understanding.

Computer programs are written in one or more programming languages, like C++, Java, or Visual Basic. A computer cannot directly understand the programming languages used to create computer programs, so the program code must be compiled. Once a program's code is compiled, the computer can understand it because the program's code is turned into machine language.

Example:

```
01001000 01100101 01101100 01101100 01101111 00100000  
01010111 01101111 01110010 01101100 01100100
```

2) Assembly language:

Type of low-level computer programming language consisting mostly of symbolic equivalents of a particular computer's machine language.

Computers produced by different manufacturers have different machine languages and require different assemblers and assembly languages. Some assembly languages can be used to convert the code that programmers write (source code) into machine language (readable by the computer), and have functions to facilitate programming (e.g., by combining a sequence of several instructions into one entity). Programming in assembly languages requires extensive knowledge of computer architecture.

Example:

Assembly Language

3

- In assembly language, a mnemonic (i.e. memory aid) is used as a short notation for the instruction to be used.

Assembly Language	Machine Code
SUB AX,BX	001010111000011
MOV CX,AX	100010111001000
MOV DX,0	101110100000000000000000

Assembly language is an intermediate step between high level languages and machine code. Most features present in HLL are not present in Assembly Language as type checking etc.

3) High-level language:

A high-level language is a programming language designed to simplify computer programming. It is "high-level" since it is several steps removed from the actual code run on a computer's processor. High-level source code contains easy-to-read syntax that is later converted into a low-level language, which can be recognized and run by a specific CPU.

Example:

- 1) C++
- 2) Java
- 3) Python

Answer 3 (b): There are two types of translator that translate high level programming language into machine language:

1) Compiler:

Compiler, Computer software that translates (compiles) source code written in a high-level language (e.g., C++) into a set of machine-language instructions that can be understood by a digital computer's CPU. Compilers are very large programs, with error-checking and other abilities. Some compilers translate high-level language into an intermediate assembly language, which is then translated (assembled) into machine code by an assembly program or assembler. Other compilers generate machine language directly.

2) Interpreter:

A high-level programming language translator that translates and runs the program at the same time. It converts one program statement into machine language, executes it, and then proceeds to the next statement. This differs from regular executable programs that are presented to the computer as binary-coded instructions. Interpreted programs remain in the source language the programmer wrote in, which is human readable text.

Interpreters and compilers are similar, since they both recognize and process source code. However, a compiler does not execute the code like an interpreter does. Instead, a compiler simply converts the source code into machine code, which can be run directly by the operating system as an executable program. Interpreters bypass the compilation process and execute the code directly.