IQRA NATIONAL UNIVERSITY
PESHAWAR

NAME: SYED JUNAID ALI SHAH

ID No: 16373

PAPER: Programming Fundamentals

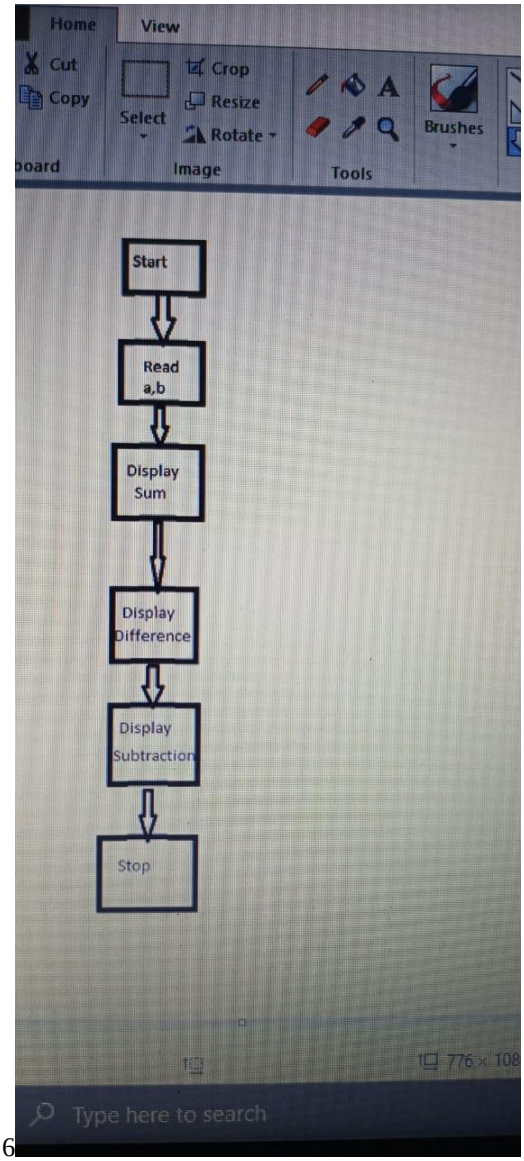
TEACHER NAME: Dr. Fazal-e-Malik

EXAM: MID TERM

Summer Semester

QNo1: a) **Draw the flow chart to get** two integer items from **keyboard** and **then** display to the screen their sum, difference and product?

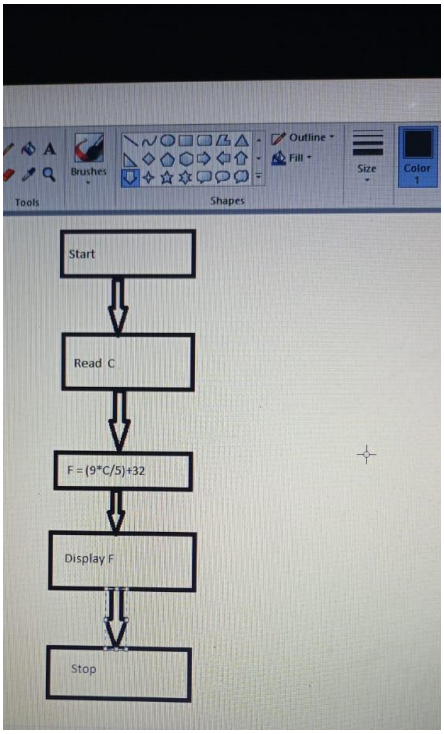
Answer:



6

QNo1: b) **Draw the flow chart to** prompt the user for a temperature in degrees Celsius (C), then convert the temperature in degrees Fahrenheit (F) using the following formula and display temperature in Fahrenheit (F) on monitor?

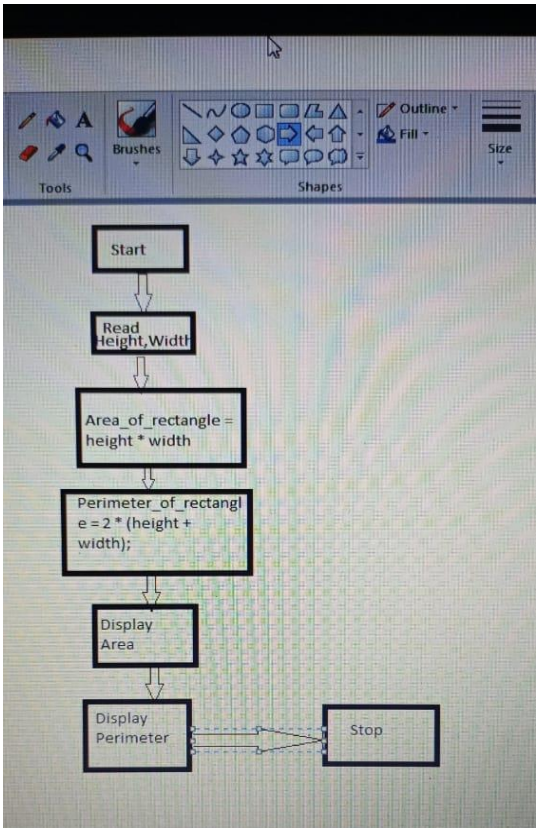
$$F = \frac{9}{5} \times C + 32$$

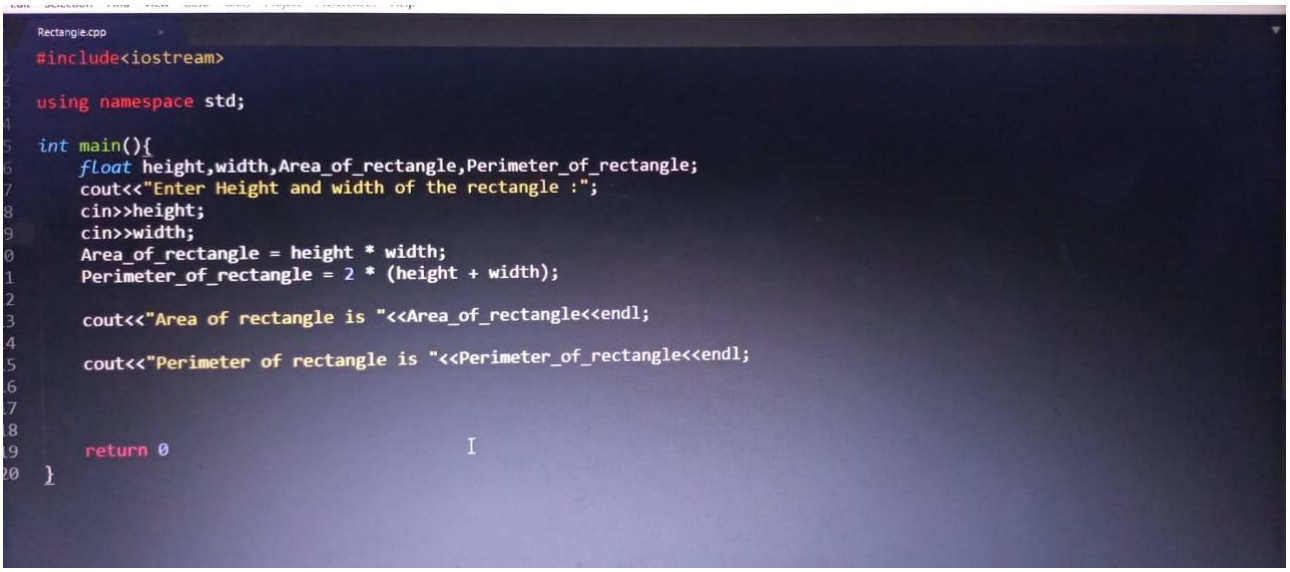


QNo2: a) **Draw the flow chart and write a C++ program** to find the Area and Perimeter of a Rectangle using the below formulae

Area of rectangle: **height*width**
Perimeter of rectangle: **2*(height + width)**

Answer:



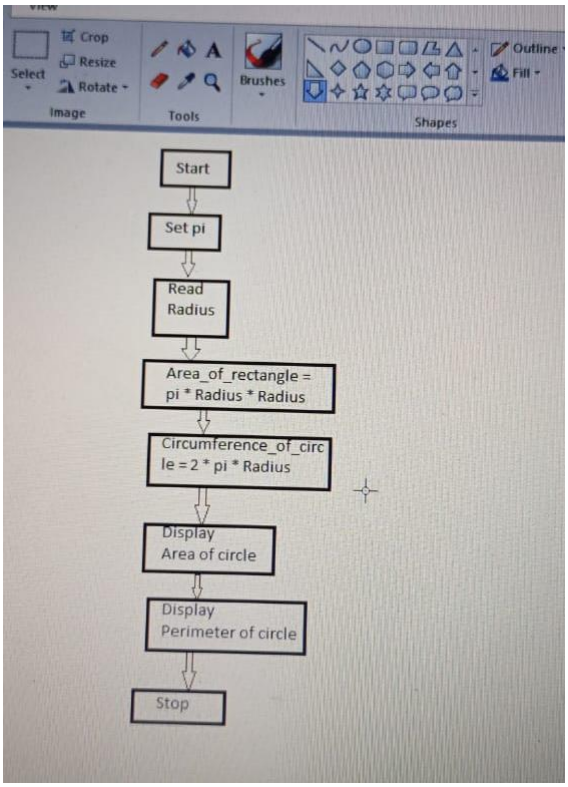


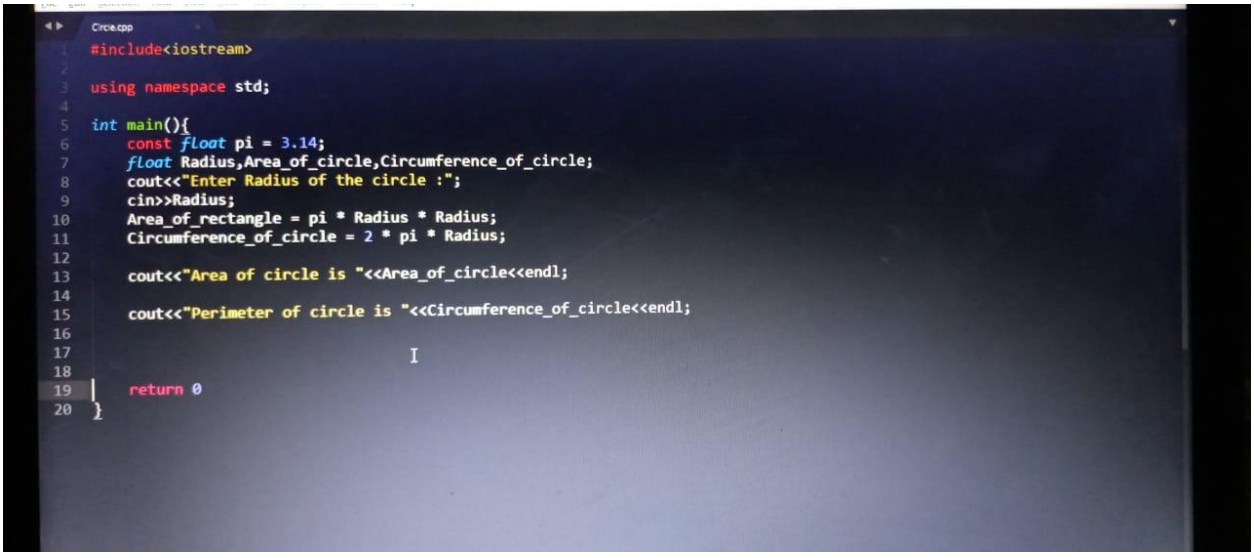
QNo2: b) **Draw the flow chart and write a C++ program to** obtain the radius of a circle. Then program calculates the area and perimeter using the below Formulae

Area of Circle = π*R*R

Circumference formula C = 2 * π * R. Where π=3.14

Answer:





```cpp
#include<iostream>

using namespace std;

int main(){
    const float pi = 3.14;
    float Radius,Area_of_circle,Circumference_of_circle;
    cout<<"Enter Radius of the circle :";
    cin>>Radius;
    Area_of_rectangle = pi * Radius * Radius;
    Circumference_of_circle = 2 * pi * Radius;

    cout<<"Area of circle is "<<Area_of_circle<<endl;

    cout<<"Perimeter of circle is "<<Circumference_of_circle<<endl;


    return 0
}
```

QNo3: a) Discuss different types of programming languages?

Answer: **Different types of programming languages:**

Programming Languages:

A programming language is a vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks.

• Programming languages can be used to create computer programs.

• The term programming language usually refers to high-level languages, such as BASIC, C, C++, COBOL, FORTRAN, Ada, and Pascal.

Computer Program

• A program is a set of instructions following the rules of the chosen language

. • Without programs, computers are useless.

• A program is like a recipe.

• It contains a list of ingredients (called variables) and a list of directions (called statements) that tell the computer what to do with the variables.

Types of Programming Language:

• There are three types of programming language:

– Machine language (Low-level language)

– Assembly language (Low-level language)

– High-level language

• Low-level languages are closer to the language used by a computer, while high-level languages are closer to human languages.

Machine Language:

- Machine language is a collection of binary digits or bits that the computer reads and interprets.
- Machine languages are the only languages understood by computers.
- While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers.
- 169 1 160 0 153 0 128 153 0 129 153 130 153 0 131 200 208 241 96
- **Example:**
- Let us say that an electric toothbrush has a processor and main memory.
- The processor can rotate the bristles left and right, and can check the on/off switch.

- The machine instructions are one byte long, and correspond to the following machine operations:

Machine Instruction	Machine Operation
0000 0000	Stop
0000 0001	Rotate bristles left
0000 0010	Rotate bristles right
0000 0100	Go back to start of program
0000 1000	Skip next instruction if switch is off

Assembly Language:

A program written in assembly language consists of a series of instructions mnemonics that correspond to a stream of executable instructions, when translated by an assembler that can be loaded into memory and executed.

• Assembly languages use keywords and symbols, much like English, to form a programming language but at the same time introduce a new problem.

The problem is that the computer doesn't understand the assembly code, so we need a way to convert it to machine code, which the computer does understand.

• Assembly language programs are translated into machine language by a program called an assembler.

Example:

– Machine language:

10110000 01100001

– Assembly language:

mov a1, #061h

– Meaning:

Move the hexadecimal value 61 (97 decimal) into the processor register named "a1".

High Level Language:

• High-level languages allow us to write computer code using instructions resembling everyday spoken language (for example: print, if, while) which are then translated into machine language to be executed.

• Programs written in a high-level language need to be translated into machine language before they can be executed.

• Some programming languages use a compiler to perform this translation and others use an interpreter.

Examples of High-level Language:

• ADA

• C

• C++

• JAVA

• BASIC

• COBOL

• PASCAL

• PHYTON

QNO3: b) How many translators are there to translate higher level language to machine language? Discuss

Answer: **Translators:**

Computers only understand machine code (binary), this is an issue because programmers prefer to use a variety of high and low-level programming languages instead.

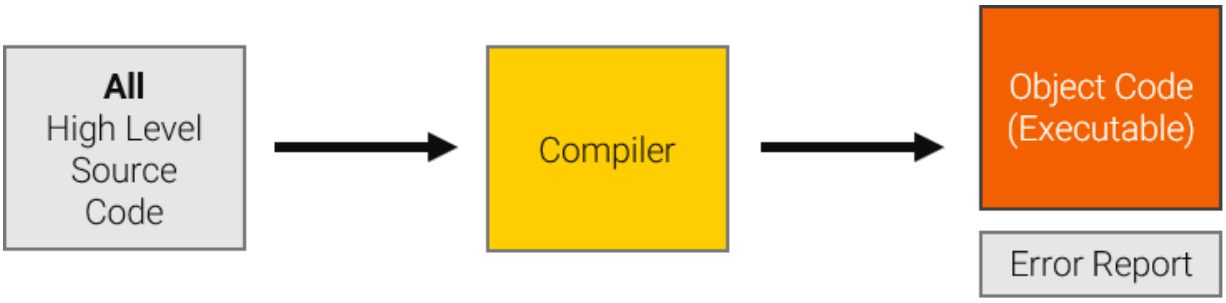
To get around the issue, the high-level and low-level program code (source code) needs to pass through a translator.

A translator will convert the source code into machine code (object code).

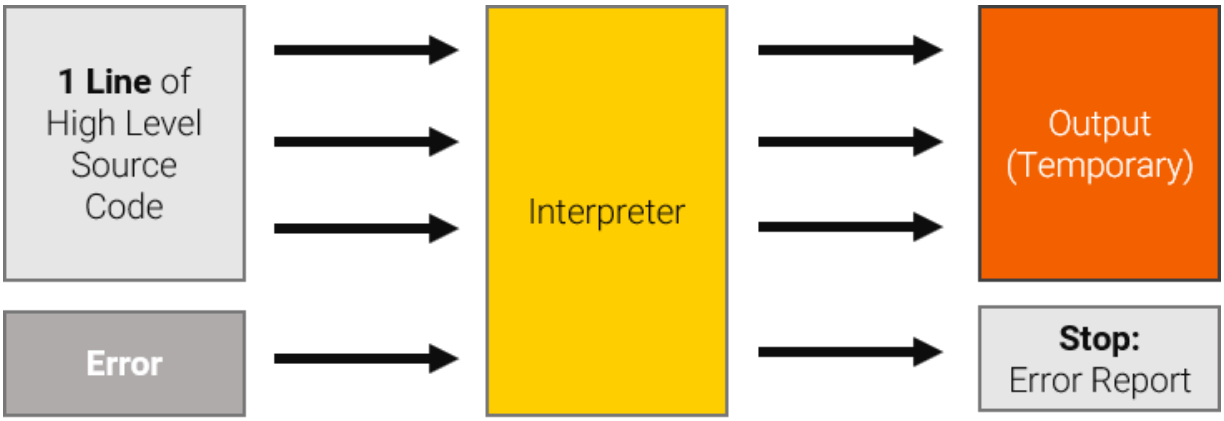
There are several types of translator programs, each able to perform different tasks.

Compiler:

A Compiler is a computer program that translates code written in a high level language to a lower level language, object/machine code. The most common reason for translating source code is to create an executable program (converting from a high level language into machine language).



Interpreter:



Interpreter programs are able to read, translate and execute one statement at a time from a high-level language program.

The interpreter stops when a line of code is reached that contains an error.

Interpreters are often used during the development of a program. They make debugging easier as each line of code is analysed and checked before execution.

Interpreted programs will launch immediately, but your program may run slower then a complied file.

No executable file is produced. The program is interpreted again from scratch every time you launch it.

Assembler:

Assemblers are used to translate a program written in a low-level assembly language into a machine code (object code) file so it can be used and executed by the computer.

Once assembled, the program file can be used again and again without re-assembly.