Iqra National University Peshawar Pakistan

Department of Computer Science

Summer Semester, Mid-Assignment, August 2020

Name	**HOORIA KHAN ORAKZAI**
ID	**14263**
Subject	**PROGRAMMING FUNDAMENTAL**
SUBMITTED TO	**SIR Fazal-e-Malik**
Department	**BS (SOFTWARE ENGINEERIN)**

Subject:	**Programming Fundamentals**	Issue Date and Time:	**25/August/2020, 9:00 am**
Program:	**BS (CS & SE)**	Submission Date and time:	**25/August/2020, 1:00 pm**
Teacher Name:	**Dr. Fazal-e-Malik**	Total Marks	**30**

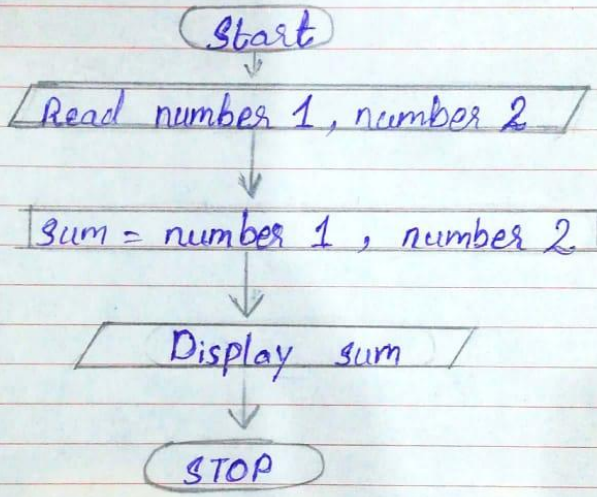
Note: Attempt all Questions. Help can be taken from net where ever is required.

Q.1

a) **Draw the flow chart to get** two integer items from **keyboard** and **then** display to the screen their sum, difference and product. 5

ID: 14263
Name: Hooria khan Orakzai
SUBJECT: Programming Fundamental.

Question No 1: (Part a)
FLOW CHART:

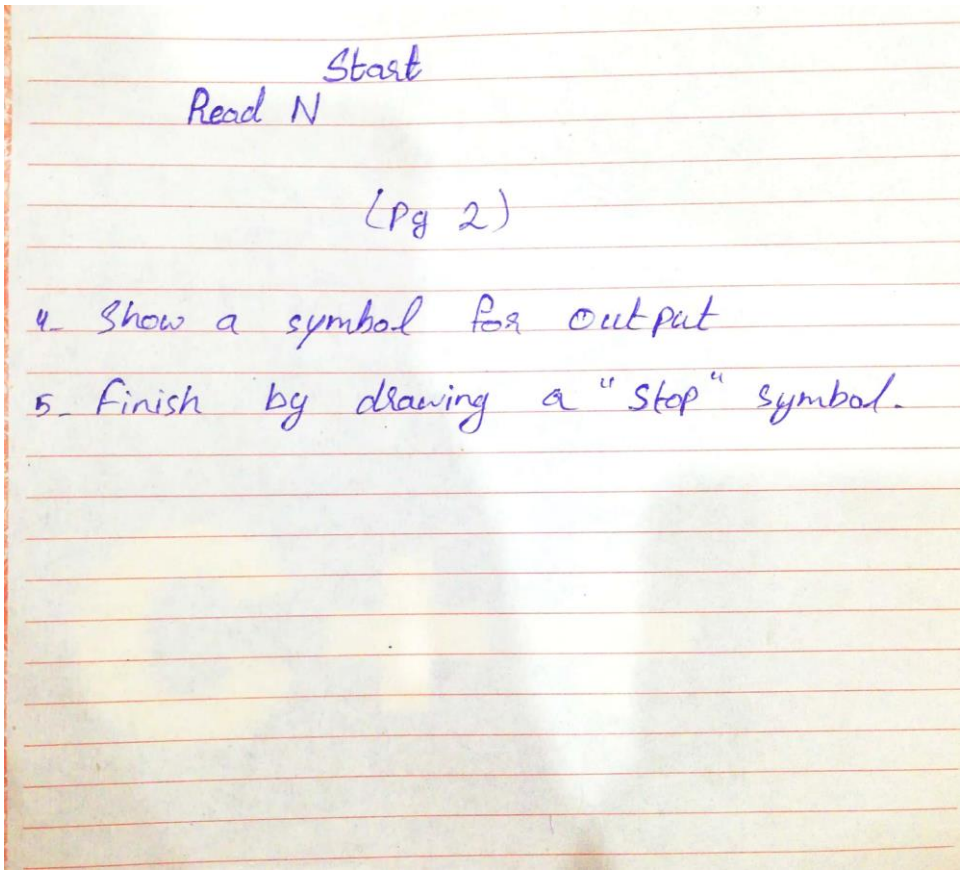


1- Start by drawing a "start" symbol.

2- Show a symbol for input.

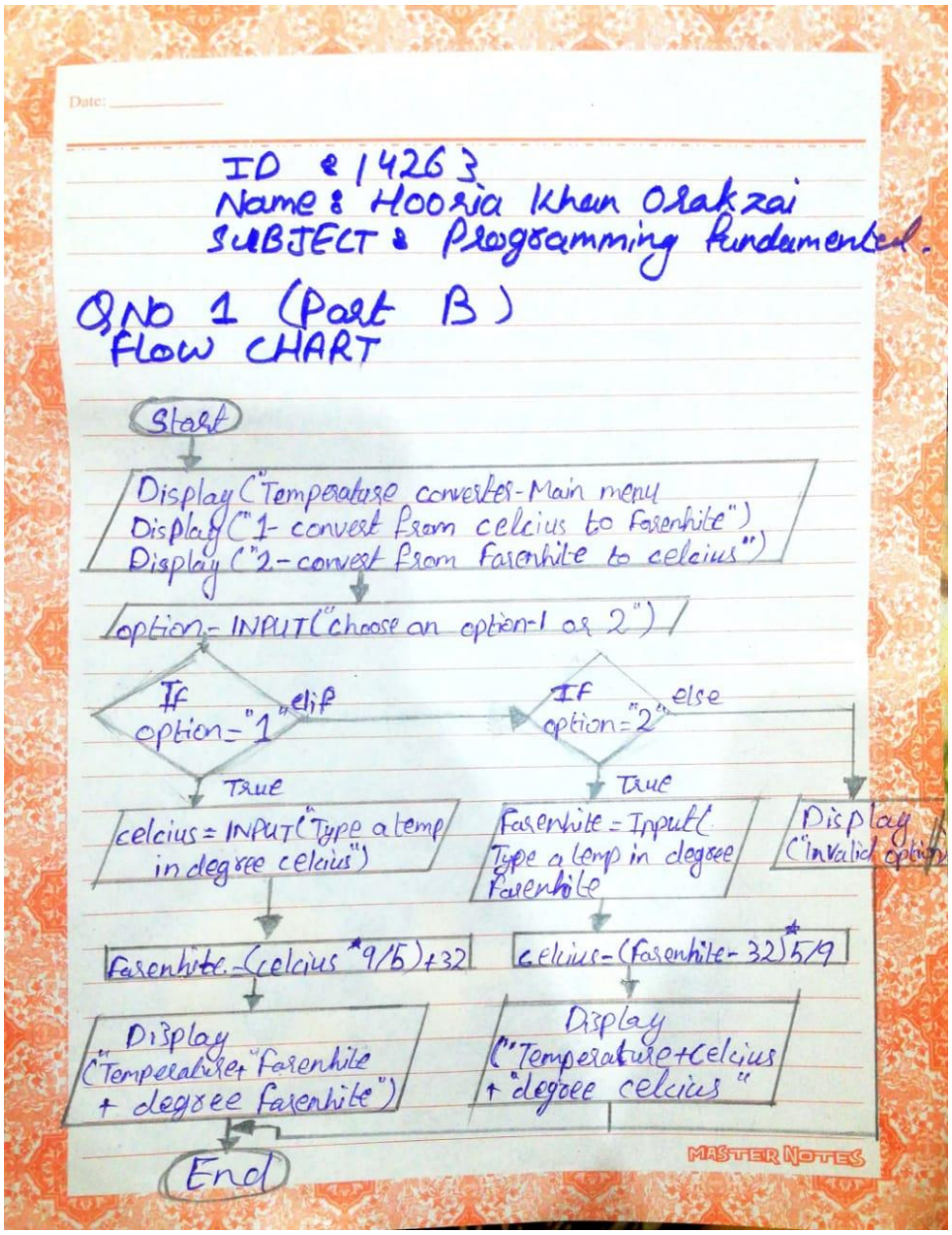
3- Show the process symbol(s) for processing.
* There can be more than 1 such steps depending on the complexity of the problem.

MASTER NOTES



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. 5

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

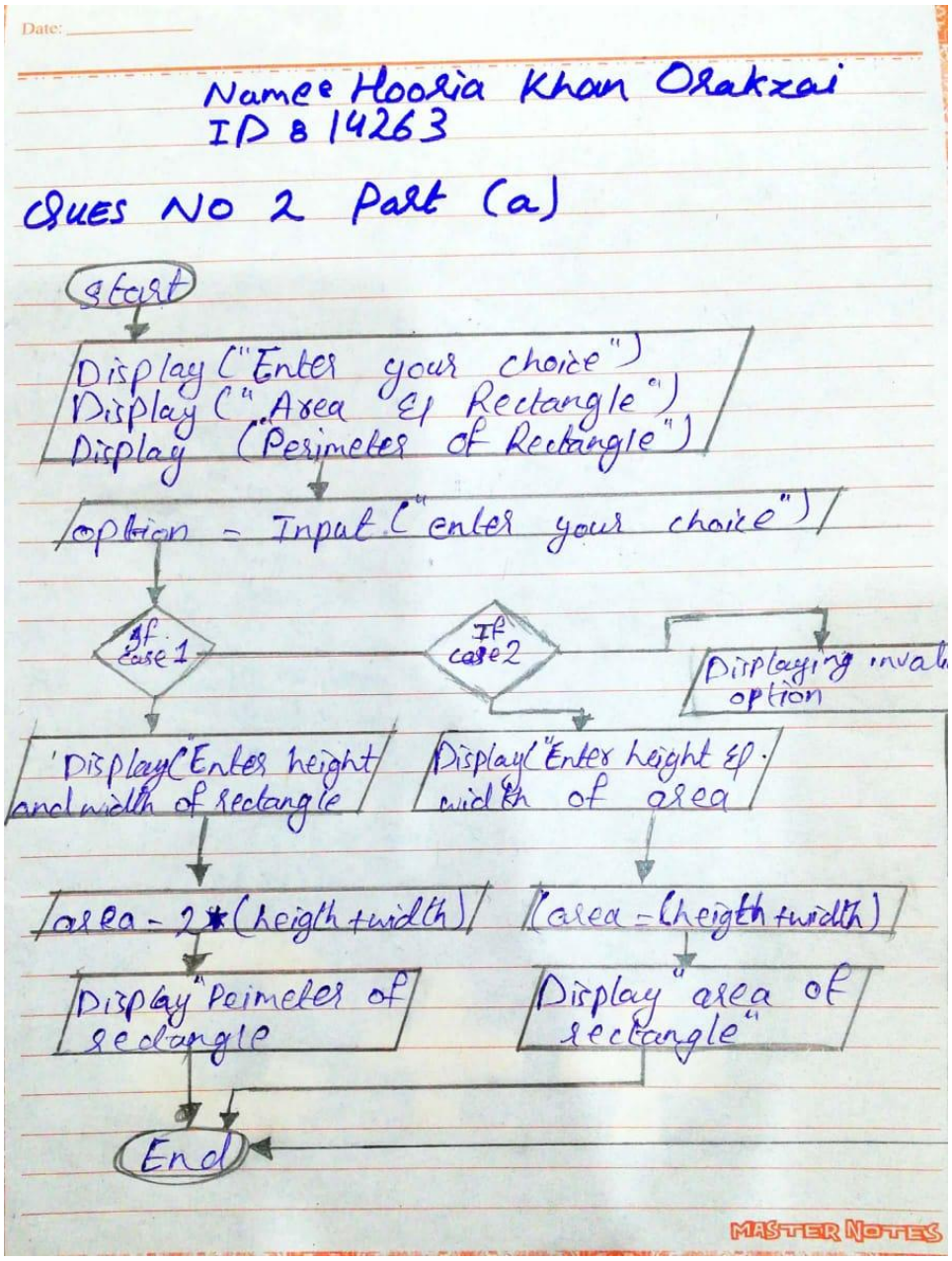


Q.2 a) **Draw the flow chart and write a C++ program** to find the Area and Perimeter of a Rectangle using the below formulae

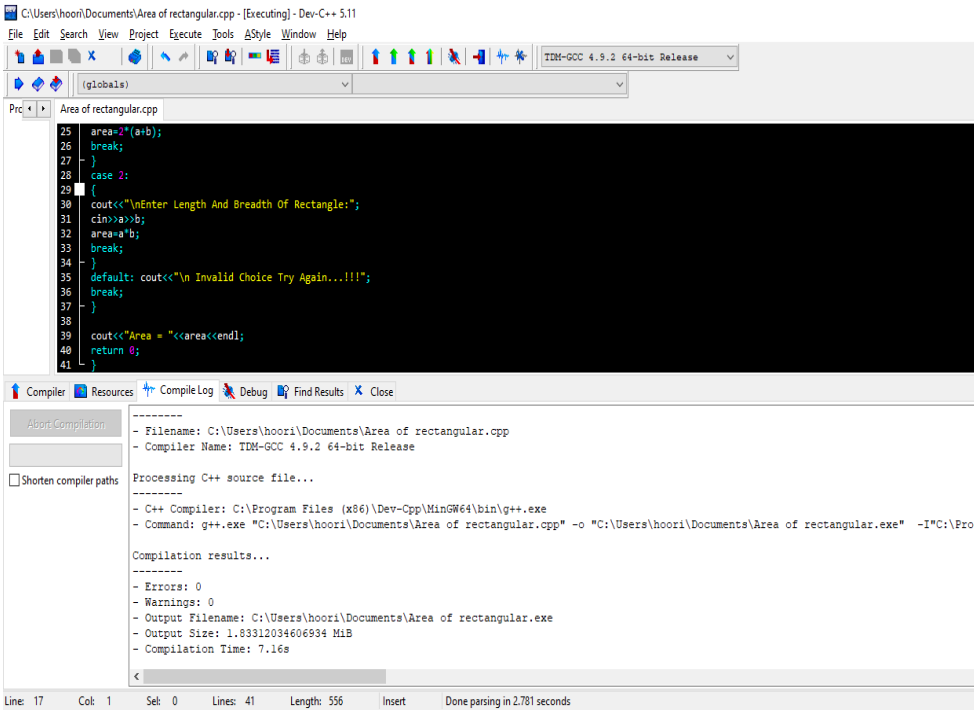
Area of rectangle: **height*width**

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

Flow Chart



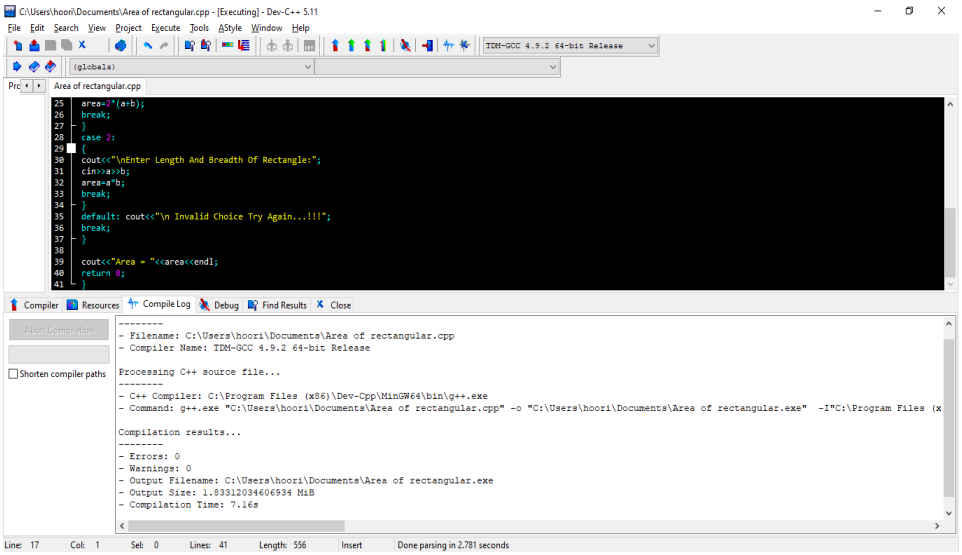
Program to find area and perimeter of the rectangle:


C:\Users\hoori\Documents\Area of rectangular.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Area of rectangular.cpp
25 area=2*(a+b);
26 break;
27 }
28 case 2:
29 {
30 cout<<"\nEnter Length And Breadth Of Rectangle:";
31 cin>>a>>b;
32 area=a*b;
33 break;
34 }
35 default: cout<<"\n Invalid Choice Try Again...!!!";
36 break;
37 }
38
39 cout<<"Area = "<<area<<endl;
40 return 0;
41 }
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths

- Filename: C:\Users\hoori\Documents\Area of rectangular.cpp
- Compiler Name: TDM-GCC 4.9.2 64-bit Release
Processing C++ source file...

- C++ Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\g++.exe
- Command: g++.exe "C:\Users\hoori\Documents\Area of rectangular.cpp" -o "C:\Users\hoori\Documents\Area of rectangular.exe" -I"C:\Pro
Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hoori\Documents\Area of rectangular.exe
- Output Size: 1.83312034606934 MiB
- Compilation Time: 7.16s
Line: 17 Col: 1 Sel: 0 Lines: 41 Length: 556 Insert Done parsing in 2.781 seconds

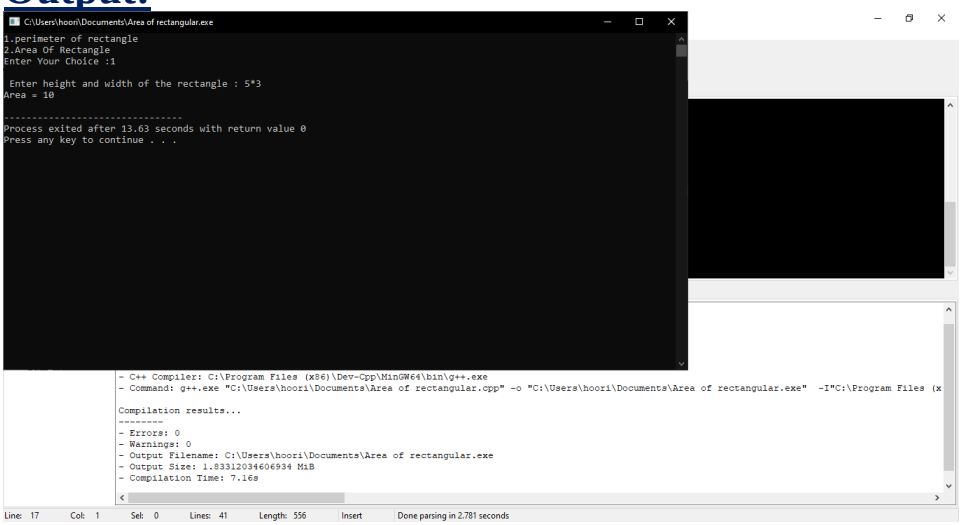

C:\Users\hoori\Documents\Area of rectangular.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Area of rectangular.cpp
25 area=2*(a+b);
26 break;
27 }
28 case 2:
29 {
30 cout<<"\nEnter Length And Breadth Of Rectangle:";
31 cin>>a>>b;
32 area=a*b;
33 break;
34 }
35 default: cout<<"\n Invalid Choice Try Again...!!!";
36 break;
37 }
38
39 cout<<"Area = "<<area<<endl;
40 return 0;
41 }
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths

- Filename: C:\Users\hoori\Documents\Area of rectangular.cpp
- Compiler Name: TDM-GCC 4.9.2 64-bit Release
Processing C++ source file...

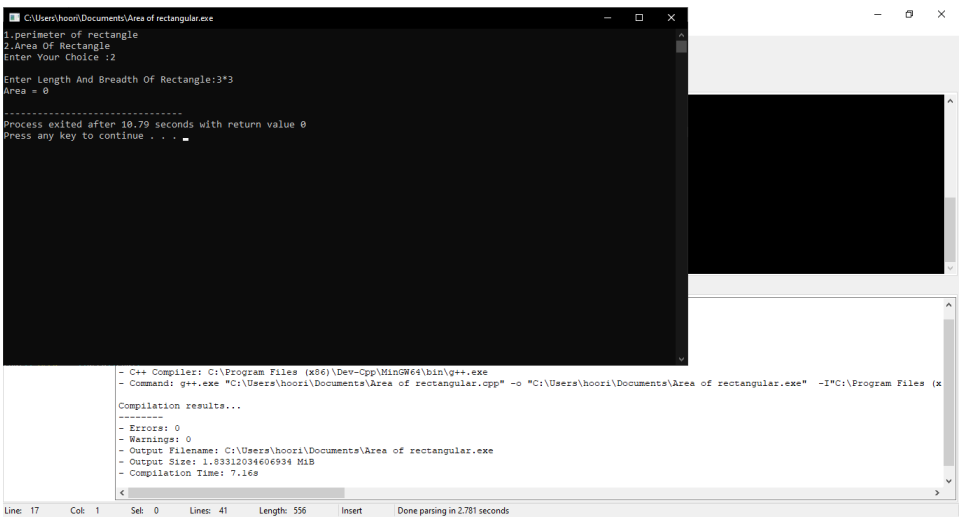
- C++ Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\g++.exe
- Command: g++.exe "C:\Users\hoori\Documents\Area of rectangular.cpp" -o "C:\Users\hoori\Documents\Area of rectangular.exe" -I"C:\Program Files (x
Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hoori\Documents\Area of rectangular.exe
- Output Size: 1.83312034606934 MiB
- Compilation Time: 7.16s
Line: 17 Col: 1 Sel: 0 Lines: 41 Length: 556 Insert Done parsing in 2.781 seconds

Output:


C:\Users\hoori\Documents\Area of rectangular.exe
1.perimeter of rectangle
2.Area Of Rectangle
Enter Your Choice :1
Enter height and width of the rectangle : 5*3
Area = 10
Process exited after 13.63 seconds with return value 0
Press any key to continue . . .
- C++ Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\g++.exe
- Command: g++.exe "C:\Users\hoori\Documents\Area of rectangular.cpp" -o "C:\Users\hoori\Documents\Area of rectangular.exe" -I"C:\Program Files (x
Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hoori\Documents\Area of rectangular.exe
- Output Size: 1.83312034606934 MiB
- Compilation Time: 7.16s
Line: 17 Col: 1 Sel: 0 Lines: 41 Length: 556 Insert Done parsing in 2.781 seconds

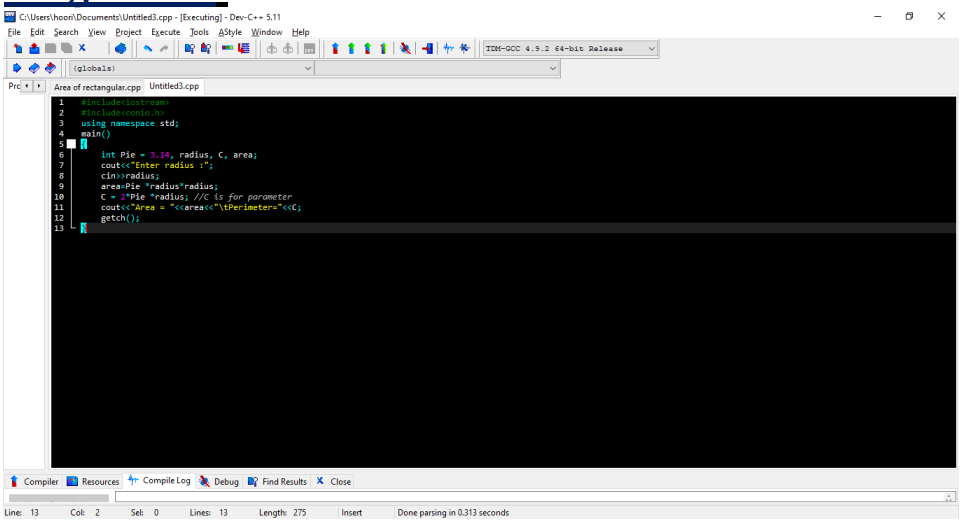


Draw the flow chart and write a C++ program to obtain the radius of a circle. Then program calculates the area and perimeter using the below Formulae

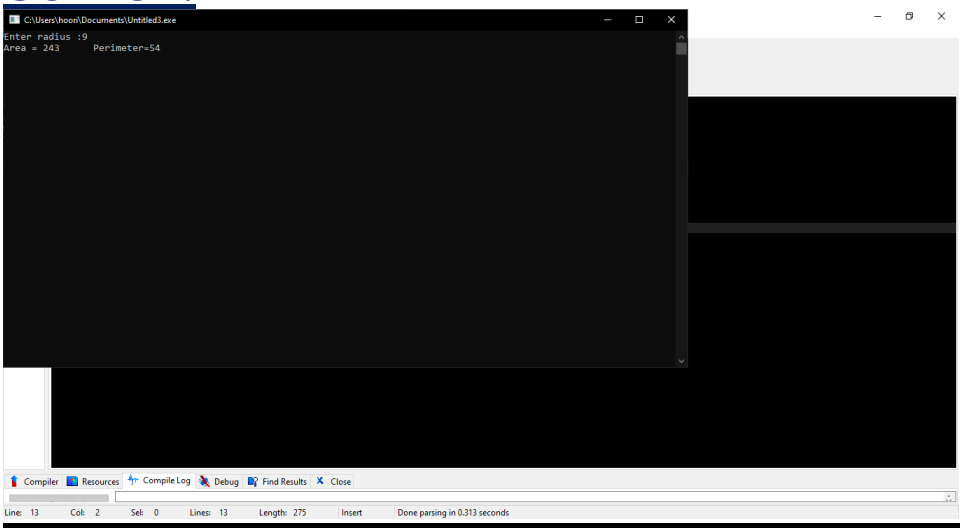
Area of Circle = π*R*R

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

Programme:



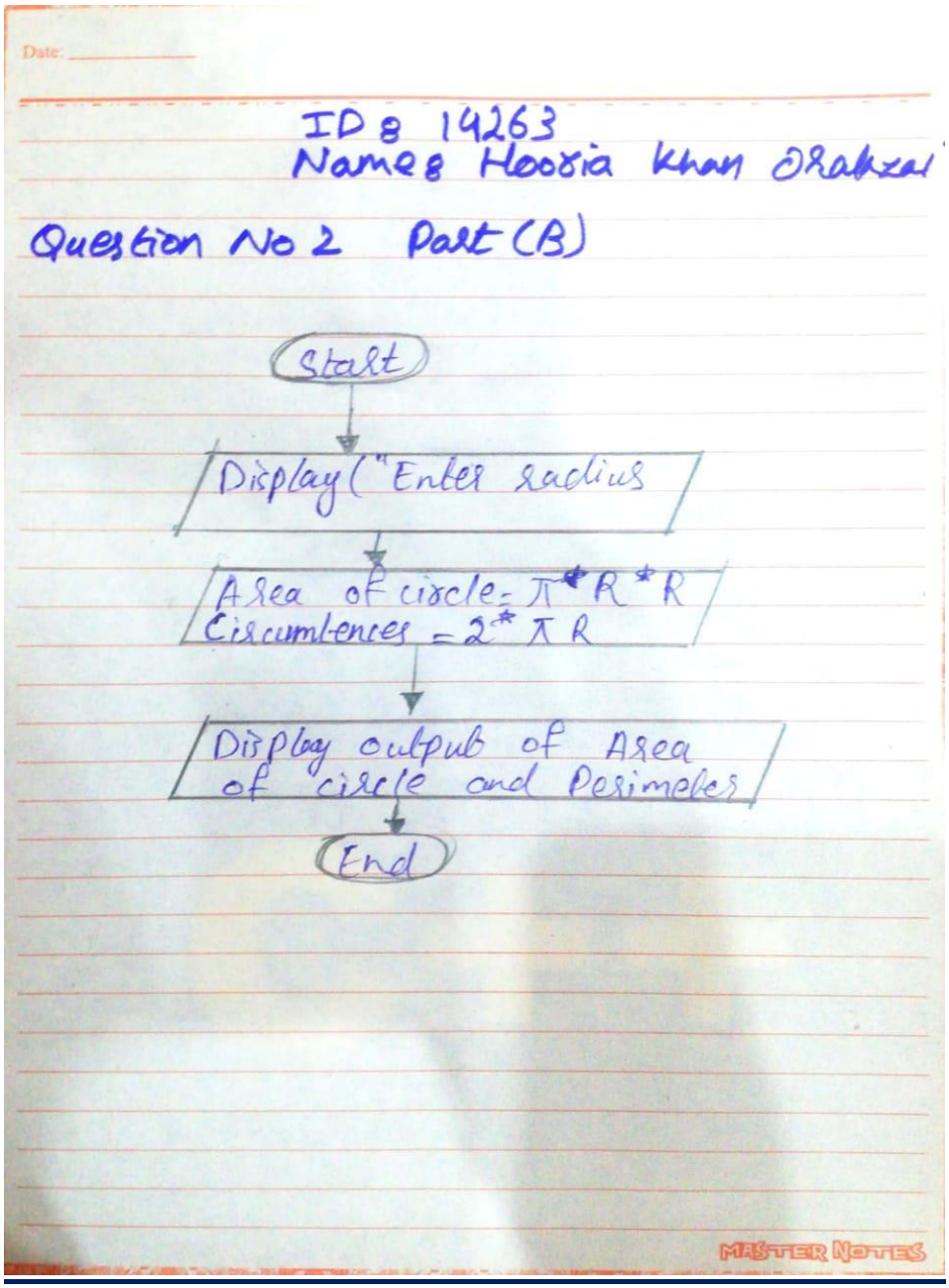
OUTPUT:



5

FLOWCHART:

5



Q.3 a) Discuss different types of programming languages.

Answer:

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

Low Level languages are further divided in to Machine language and Assembly language

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

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

Answer:

There are two types of translator

1. Compiler
2. Interpreter

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

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.

☺ **********☺

Note: Please write your Name and ID on top of your answer Paper otherwise you will get zero marks.