1.

#include<stdio.h>

Int main()

{

//program to store and manage students records

Char RegNumber[20], FirstName[10],LastName[10];

Float IntProg, NetWrk, Acc, OS, CompApp;

printf(“\nEnter RegNumber: ”);

scanf(“%s”, &RegNumber);

printf(“\nEnter FirstName: ”);

scanf(“%s”,&FirstName);

printf(“\nEnter LastName: ”);

scanf(“%s”,&LastName);

printf(“\nEnter IntroProg: ”);

scanf(“%f”,IntroProg);

printf(“\nEnter NetWrk: ”);

scanf(“%f”,NetWrk);

printf(“\nEnter Acc: ”);

scanf(“%f”,Acc);

printf(“Enter OS: ”);

scanf(“%f”,OS);

printf(“\nEnter CompApp: ”);

scanf(“%f”,CompApp);

printf(“\nStudent Records”);

printf(“\nRegistration Number is %s”,RegNumber);

printf(“\nFirst Name is %s”,FirstName);

printf(“\nLast Name is %s”,LastName);

printf(“\nIntroduction To Programming Score is %f”,IntroProg);

printf(“\nNetworking Score is%f”,NetWrk);

printf(“\nAccounting Score is%f”,Acc);

printf(“\nOperating System Score is %f”,OS);

printf(“\nComputer Application Score is %f”,CompApp);

return 0;

}

2.

* Compiler

It is a language translating program that converts source code into machine code before the written code of program is run.

* Source code

They are programming statements in human readable form saved as a file yet to be converted into binary language.

* Object code

It is a compiled file that has been converted from the human readable source code to machine language for execution by the processor.

* Linkers

This is a utility program that joins object files produced by the compiler and other code to form a single executable file.

3.

#include <stdio.h>

int main()

{

//Creating a program that finds sum of two numbers

double numb1=12.453, numb2=14.579, sum;

printf("Enter two numbers:");

scanf("%.2lf%.2lf",&numb1, &numb2);

sum=numb1+numb2;

printf(" Sum = %.2lf", sum);

return 0 ;

}

From the program above, variables of data type double have been created to store literal values, as per the variable names. The first printf () function prompts the user to assign values to the values to the variables ie.printf (“Enter the two Numbers :”);

The scanf () function reads in the entered values into the specified variables.

The values stored in numb1 and numb2 variables are then added up and the result stored in the sum variable ie. Sum = numb1+numb2

The last printf () function displays the value stored in the sum variable on the screen.

4.

|  |  |
| --- | --- |
| **COMPILER** | **INTERPRETER** |
| 1. Translates the entire source code at once then executes the code. | 1. Translates the source code and executes one line at a time. |
| 2. Fast execution since object code is saved on a storage medium and is fetched anytime the program runs. | 2. Slow execution since the program is translated each time it is run. |
| 3. Saves object code in memory hence requiring more space. | 3. Does not save object code in the memory hence requires less memory. |
| 4. It is not possible to partially test and debug the code during development. | 4.Possible to partially test and debug the code during development |
| 5. Program need not to be compiled every time it is time. | 5. The source code is converted to object code every time it is run. |
| 6. Errors are displayed after entire program is checked. | 6. Errors are displayed for every instruction code interpreted (if any). |
| 7.Example:C compiler | 7.Example:BASIC |

5.

**Operators**

Based on their priority of execution by the processor;

1. Arithmetic Operators

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulus |

1. Relational Operators

|  |  |
| --- | --- |
| **Operator** | **Meaning** |
| < | Less Than |
| > | Greater Than |
| <== | Less Than or Equal To |
| >== | Greater Than or Equal To |
| == | Equal To |
| != | Not Equal To |

1. Logical Operators

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| && | Logical AND |
| || | Logical OR |
| ! | Logical NOT |

1. Assignment Operators

It is denoted by ’=’

Syntax: variable=expression; or variable= value; eg. int numb=16

1. Increment/Decrement Operators

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| ++ | Increment |
| \_ \_ | Decrement |

1. Conditional Operator

It is denoted by ‘?’

Syntax;(condition)?exp1:exp2;

* If condition is true,exp1 is executed.
* If condition is false,exp2 is executed.

Eg.x=(y>0)?1:0;