1. Define the following terms as used in C programming;
2. Compiler; it is a program that converts instructions into lower-level or a machine-code form so that they can be read and executed by a computer.
3. Source code; A text listing of commands to be complied or assembled into an executable computer program.
4. Object code; Is a code produced by a compiler or assembler.
5. Linker; A program used with a compiler to provide links to the libraries needed for an executable program.
6. Using an example, i.e , a program to add two numbers ; explain the compilation process of a C program.

Answer;

Printf(“ Enter two inters:”);

Scanf(“ %d%d”,&number1,&number2);Then, these two numbers are added using the + operator , and the result is stored in the sum variable. Finally, the printf () function is used to display the sum of numbers. Printf(“ %d+%d”, number 1, number 2,sum);

Example;

#include<stdio.h>

Int main ()

{

//Declare variables

Int num1,num2,sum;

Printf( “Enter two integers:”);

Scanf(“%d%d”),&num1 ,&num2);

//calculate the sum of the two numbers.

4.Explain the differences between a compiler and an interpreter (atleast 6 comparisons)

|  |  |
| --- | --- |
| COMPILER | INTERPRETER |
| 1. Takes in the entire program and requires a lot of time to analyze the source code. | 1. Takes a single line of code and very little time to analyze it. |
| 1. Compiled code runs faster | 1. Interpreted code runs slower |
| 1. A compiler displays all errors after compilation. If your code has mistakes , it will not compile. | 1. The interpreter displays errors of each line one by one. |
| 1. A compiler transform high level language into machine language. | 1. An interpreter is a computer software that runs code written in a high level language. |
| 1. The compiler generates an output in the form of (exe) | 1. The interpreter does not generate any output |
| 1. It does not require a source code for later execution | 1. It requires source code for later execution |

5) List all the categories of operator available in C programming and specific operators;

|  |  |
| --- | --- |
| TYPES OF OPERATORS | SPECIFIC OPERATORS |
| Increment and decrement | Increment (++), decrement (-) |
| Bitwise | Bitwise AND, Bitwise OR, Bitwise LEFT SHIFT, Bitwise XOR, Bitwise ONE’S COMPLEMENT |
| Assignment | = , += ,-= , /= , \*= , %= |
| Logical | && , || , ! |
| Relational | == , != , <= , >= , < ,> |
| Special | & , \* , Sizeof |
| Conditional | != ,&gt , &lt ,&gt;= |
| arithmetic | +, \*, - , /,% |