Tutorial 01

Q1) A programming language is essential because it gives people an organized, standardized way to give instructions to machines. It acts as a link between what people think and what machines do. Programmers can solve complex issues, automate operations, and create software applications by using a programming language to construct accessible and intelligible code. The syntax and rules of the language ensure that the computer interprets the instructions correctly, enabling efficient and precise execution of the required operations.

Q2)

i)Source Code vs. Machine Code

Computer programmes can be represented by either source code or machine code. A high-level programming language, like C or Python, is used to create source code, which is legible by humans. It is composed of sentences and directives that follow the syntax and regulations of the programming language. A compiler or interpreter receives source code as input. Machine code, on the other hand, is a binary representation of instructions that can be carried out instantly by a computer's hardware. It is unique to the architecture of the computer and consists of a string of 0s and 1s. The compilation procedure converts the source code into an understandable and executable representation known as machine code. Machine code, in contrast to source code, is difficult to read.

ii)High Level Language vs. Low Level Language

Programming languages fall into two major categories: high-level languages and low-level languages, each with its own traits and objectives. High-level languages are intended to be more user-friendly and abstract from the underlying hardware, examples of which include Python, Java, and C++. They offer more advanced building blocks like variables, functions, and objects that make programming simpler and more effective. Because high-level languages are frequently easier to read and maintain, developers may concentrate on finding solutions rather than worrying about tiny details. Low-level languages, such as assembly language, on the other hand, are more closely related to the hardware and offer more direct access to the system's resources. They require a thorough understanding of the hardware architecture since they involve working with specific instructions and memory addresses.

iii)Compiler vs. Interpreter
There are two different ways to run code created in a programming language: a compiler and an interpreter. Before being executed, a compiler converts the complete source code into machine code or an executable file. It carries out analysis, makes optimisations, and produces output that can be used independently. Instead of creating an executable file, an interpreter examines and runs the source code line by line. It directly understands and executes the instructions, giving you more flexibility but often executing more slowly than compiled code.

iv)Structured Language vs. Object Oriented Language
Code is organised into modular structures using control flow components like loops and conditionals in structured programming languages like C. They promote a procedural approach to programming by separating data and behaviour. Languages that are object-oriented, like Java and C++, place more emphasis on the idea of objects and classes. They enable modular and reusable code by encapsulating data and behaviour into objects. Compared to structured languages, OOP languages offer a more modular and flexible approach to software design and development by supporting notions like inheritance, polymorphism, and encapsulation.

v)C vs C++
Programming languages C and C++ both have their advantages and disadvantages. The procedural programming language C is renowned for being straightforward, effective, and capable of low-level system programming. It offers fundamental data kinds, control schemes, and operations. Contrarily, C++ is an expansion of C that includes additional functionality for object-oriented programming. While adding classes, objects, inheritance, and other OOP ideas, C's features are preserved. For more complex programming jobs, C++ also provides extra capabilities including templates, exception handling, and the Standard Template Library (STL).

vi)C++ vs Java
Java and C++ are both widely used programming languages, although they have some key distinctions. The statically typed language C++ enables greater direct control over hardware and memory. Operator overloading, manual memory management, and low-level programming are all supported. Java, on the other

hand, uses automatic memory management and is a garbage-collected language. It emphasises security, allows multithreading, and is platform-independent. Java is a good choice for developing enterprise apps and websites since it has built-in support for networking and GUI development.

vii)Syntax error vs Logical error
When code deviates from the grammar and rules of the computer language, a compilation error results, which is known as a syntax error. Semicolons that are absent or have the wrong brackets are two examples. On the other hand, a logical error is a fault in the logic of the programme that results in it producing unexpected or wrong outcomes.

TUTORIAL 02

1) The goal of comments in a program is "to give the programmer an idea of what the program says." Comments are written in C programs by beginning them with
//.

2) The primary purpose.

3. To get user inputs.

4) • Yes. The language has case sensitivities.

5)Record 1, tax, name, and name_and_address are all valid values.

• Invalid: name-and-address, return, 1 record, file-3, and 123-45-6789 Because identifiers cannot begin with a number, 1record is invalid.

o File-3 is invalid since hyphens or minus signs are prohibited in identifiers. Because return is a reserved keyword in C, it is invalid.

A space cannot be used in an identifier, hence the name and address are invalid.

Name-and-address is used since IDs cannot contain hyphens or other symbols.

Because hyphens and minus signs are not permitted in identifiers, the number 123-45-6789 is invalid.

6) Indicate each of the following statements as true or false. If false, why not?

A. The printf function always starts printing at the start of a new line. - false

b. When the program is run, comments force the computer to print the text enclosed between /* and */ on the screen.

- false

c. The cursor positions to the start of the next line on the screen when the escape character n is used in a printf format control string. – true

d. Prior to usage, all variables must be defined. - true

e. When a variable is defined, it must be given a type.

The variables number and number are seen as being the same by f. C. - false

g. There must be three printf instructions in a program that prints three lines of output. – false

```
7)*
  **
  ***
  ****
  *****
```

8)

Scanf("d", value); - There are no% and & before d and value, respectively.

b. printf("The product of%d and%d is%d"n, x, y);- There are only 2 variables but 3%d symbols, and the n is written outside the "" symbol.

C. There is no & before anlnteger in scanf("%d", anlnteger);

d. printf("Remainder of%d divided by%d is n", x, y, x%y);- There are three variables, but only 2%d symbols.

e. The comma is inside the "" and the print command is incorrect. Print("The sum is%dn," x+y); "printf" is the appropriate command.

f. Printf("The value you entered is:%dn,&value;- there is an additional & Infront of value and there is no "" symbol at the end of the value

9) When each of the following statements is run, what, if anything, prints? In that case, simply respond "Nothing." Assume x is 2 and y is 3.

g. printf(“%d”, x); - 2

h. printf(“%d”, x + x); - 4

i. printf(“x=); - x=

j. printf(“x=%d”, x); - x=2

k. printf(“%d = %d”, x + y, y + x); -5 = 5

l. z=x+y; - Nothing

m. scanf(“%d%d”, &x , &y);- Nothing

n. /*printf(“x + y = %d”, x + y); */- x+y=5

o. Printf(“\n”);- A line break will print

10) C operators are analyzed in the left to right direction. -false It depends on the order in which the operators are used.

o) The variable names listed below are all acceptable. His account total is: a, b, c, z, 2, under_bar, m928134, t5, j7, her_sales, and z2.

– true

p) An assignment statement is often expressed as printf("a=5",);.

- false

➢Because printf("a=5"); q) only applies to the most common assignment statement, a=5, a valid arithmetic expression without parenthesis is evaluated from left to right. It depends on the order in which the operators are used.

r) The variable names listed below are all incorrect. 3g, 87, 67h2, h22, 2h -false, as h22 is a recognized variable name.

TUTORIAL 03

11)

1 – int x=0; x=x + 1;

2 – int x=0; x+=1;

3 – int x=0; x++;   4 – int x; x=0+1;

12)

a) After calculating the sum of x and y, assign it to z and increase the value of x by 1.

Z=x+y;x++; b) Use the = and * operators to multiply the variable product by 2.   product*=2

c) Use the = and * operators to multiply the variable product.

Product=product*2

d) Determine whether the variable count value is more than 10. If so, write "Count is greater than 10."

If count is larger than 10, print "Count is greater than 10."

e) Decrease the variable x by 1 and then deduct it from the total variable.

X++;y=total-x;

f) Increase the variable x and decrease the variable total by the same amount. Y=x+total; x - -;

g) Determine the residual after dividing the variable q by the divisor and apply the result to the variable q. Write two versions of this sentence.

f) Increase the variable total by adding the variable x, then decrease the variable x by 1 Y=x+total; x - -;

g) Determine the remaining after dividing the variable q by the divisor and assign the result to the variable q. This sentence should be written twice.

Dividend d

First Way: Q = Q%D; Second Way: P = Q%D; Q = P;

h) Write the number 123.4567 with two digits of accuracy. What is printed as a value?

123.45

i) Print, three digits to the decimal point, the floating-point value 3.14159. What is printed as a value?

3.141

13)

A. Input the integer variable x using scanf. Scanf("%d", &x);

B. Input the integer variable y using scanf. Scanf("%d", &y);

C. Initialize the integer variable i to 1 Int = 1;

D. Initialize the integer variable power to 1 Int power = 1

E. Add the result of multiplying variable power by x to power.

Power = power*x; f. Increase variable i by 1; I=i+1;

G. Check I in the while statement's condition to see if it's less than or equal to y.

While(i<=y)

H. Use printf to output integer variable power. Power = Printf("%d",power);

TUTORIAL 04

14)

```
If numNeighbors >=3 | | numNeighbors = 4

++numNeighbors;

Printf("You are dead! \n");

Else

--numNeighbors;
```

Errors include: not using () to denote the if statement's condition, not indenting the statements inside of if and else, and placing the second condition inside the first.

15)

```
        Int number = 4;

        Double alpha = -1.0;

        If (number>0)

        If (alpha>0)

Printf("Here I am! \n");

Printf("No, actually, I'm here! \n);
```

Output: I'm actually here! Actually, I'm here, so no! 16) Take into account the following if statement, in which the variables doesSignificantWork, makesBreakthrough, and nobelPrizeCandidate are all Boolean:

Assuming (doesSignificantWork)

If(makesBreakthrough)

nominee for the Nobel Prize is true;

else

candidateForNobelPrize is false;

}

If (!doesSignificantWork) then else

candidateForNobelPrize is false;

17)

Add the taxRate % of price to price if character variable taxCode is 'T'.

```
If(taxCode=’T’){price=price+taxRate;}
```

- If the value of the integer variable opcode is 1, read in the double values for x and y, calculate their sum, and output it.

```
If (opcode=1), print ("The Sum Is%f") (x+y);
```

- If the integer variable currentNumber is even, alter its value to be half of currentNumber (rounded down when currentNumber is odd). Otherwise, modify its value to be three times currentNumber plus one.

```
     If(“currentNumber%2=1){currentNumber=3*currentNumber+1;}
Else{currentNumber=currentNumber/2;}
```

- Assign true to the Boolean variable leapYear if the integer variable year is a leap year. (A leap year is a multiple od 4, and if it is a multiple of 100, it must also be a multiple of 400.)

```
   If(year%4=0){leapYear=true;}
```

- Assign a value to double variable cost depending on the value of integer variable distance as follows:

Distance	Cost
0 through 100	5.00
More than 100 but not more than 500	8.00
More than 500 but less than 1,000	10.00
1,000 or more	12.00

If(distance<=100){cost=5.00;} else if(distance<=500){cost=8.00;} else if(distance<1000){cost=10.00;} else if{cost=12.00;}

TUTORIAL 05

18)SWITCH

Enter two numbers, and the results of the fundamental mathematical operations will be displayed. The following should be shown on the output screen:

Enter two numbers ___

1. +

2. –

3. *

4. /

Please enter your choice__

WHILE LOOP

19) Enter 10 numbers, and a total number of odd and even numbers in the number series will be displayed.

```
#include <stdio.h>

Int main ()

{

Int a, i=1, oc=0, ec=0;

While (i<=10)
```