10th Class		
Computer Science	Model Paper 1	Paper: II
Time: 1.45 Hours	(Subjective Type)	Marks: 40

(Part-I)

2. Write short answers to any FOUR (4) questions: (8)

(i) Why do we need a programming environment?

Ans It is essential to setup a programming environment before we start writing programs. It works as a basic platform for us to write and execute programs.

(ii) Differentiate between char and int.

Ans **Character - char:**

To declare character type variables in C, we use the keyword *char*. It takes up just 1 byte of memory for storage. A variable of type *char* can store one character only.

Integer - int (signed/unsigned):

Integer data type is used to store integer values (whole numbers). Integer takes up 4 bytes of memory. To declare a variable of type integer, we use the keyword *int*.

(iii) Define statement terminator.

Ans A **statement terminator** is identifier for compiler which identifies end of a statement. In C language, semicolon (;) is used as statement terminator.

(iv) What is modulus operator?

Ans **Modulus operator** is also a binary operator, which performs division of left operand to the right operand and returns the remainder value after division. Modulus operator works on integer data types.

(v) Write down output of the following code segment.

```
# include<stdio.h>
void main ()
{
        int x = 2, y = 3, z = 6;
        int ans1, ans2, ans3;
        ans1 = x / z * y;
        ans2 = y + z / y * 2;
```

```
        ans3 = z / x + x * y;
        printf("%d %d %d", ans1, ans2, ans3);
    }
```

Ans Output 0, 7, 9.

(vi) Identify errors in the following code.

```
#include<stdio.h>
void main ()
{
        int a, b = 13;
        b = a % 2;
        printf("Value of b is : %d, b);
}
```

Ans **Errors:**

1. Value of variable is not initialized.
2. Inverted comma is missing after %d.

---

3. Write short answers to any FOUR (4) questions: (8)

(i) Define Control Statements.

Ans The flow of program execution is controlled through control statements.

(ii) What are nested selection structures?

Ans Selection statements within selection statements are called **nested selection structures**.

(iii) Why do we need selection statements?

Ans Because it helps us to decide which statements should be executed next, on the basis of conditions.

(iv) Identify error in the following code segment.

```
If (x ≥ 10)
        printf ("Good");
```

Ans **Error:** Wrong use of the symbol ≥.

(v) Write down output of the following code.

```
int a = 7, b = 10;
a = a + b;
if (a > 20 && b < 20)
        b = a + b;
printf ("a = %d, b = %d", a, b);
```

Ans **Output:** a = 17, b = 10.

(vi) Is loop a data structure? Justify your answer.

Ans Yes, loop is a data structure because loop is a programming structure that repeats a sequence of instructions until a specific condition is met.

Computer programmers use loops to store data, cycle through values, add sums of numbers, repeat functions and many other things.

4. Write short answers to any FOUR (4) questions: (8)

(i) Define data structure.

Ans Data structure is a container to store collection of data items in a specific layout.

(ii) What is the use of nested loops?

Ans When we want to repeat a pattern for multiple times, then we use nested loops. *e.g.,* if 10 times we want to display the numbers from 1 – 10. We can do this by writing the code of displaying the numbers from 1 – 10 in another loop that runs 10 times.

(iii) Identify the errors in the following code.

```
int a[ ] = ({2}, {3}, {4});
```

Ans **Error:**

Parenthesis are not used in array initialization and single curly braces is used for 1d array.

(iv) Write down output of the following code.

```
int sum = 0, p;
for (p = 5; p < = 25; p = p + 5)
        sum = sum + 5;
printf ("Sum is %d", sum);
```

Ans Output = 25.

(v) Define reusability.

Ans Functions provide reusability of code. It means that whenever we need to use the functionality provided by the function, we just call the function. We do not need to write the same set of statements again and again.

(vi) Enlist the parts of a function definition.

Ans **Parts of a function definition:**

A function consists of the following parts:

1. declaration

2. function body
3. function call

Function declaration and body are mandatory, while a function call can be optional in a program.

(Part-II)

NOTE: Attempt any TWO (2) questions.

Q.5. Describe the purpose and syntax of comments in C program. (8)

Ans **Purpose and Syntax of Comments in C Programs:**

Comments are the statements in a program that are ignored by the compiler and do not get executed. Usually, comments are written in natural language *e.g.*, in English languages, in order to provide description of our code.

Purpose of writing comments:

Comments can be thought of as documentation of the program. Their purpose is twofold:

(i) They facilitate other programmers to understand our code.

(ii) They help us to understand our own code even after years of writing it.

We do not want these statements to be executed, because it may cause syntax error as the statements are written in natural language.

Syntax of writing comments:

In C programming language, there are two types of comments:

(i) Single-line Comments.

(ii) Muiti-line Comments.

Single-line comments start with //. Anything after // on the same line, is considered a comment. For example, // This is a comment.

Multi-line comments start with /* and end at */. Anything between /* and */ is considered a comment, even on multiple lines. For example,

```
/*this is
a multi-line
comment*/
```

Q.6. Write a program that takes two integers as input and tells whether first one is a factor of the second one? (8)

Ans

```
#include<stdio.h>
#include<conio.h>
int main()
{
in num1, num2;
printf("Please enter the first number: ");
scanf("%d",&num1);
printf("Please enter the second number: ");
scanf("%d", &num2);
if(num1%num2==0)
    printf("First first number is factor of second Number: ");
else
    printf("First first number is NOT factor of second Number: ");
getch();
}
```

Q.7. What is the difference between arguments and parameters? Give an example. (8)

Ans The values passed to the function are called **arguments**; whereas variables in the function definition that receive these values are called **parameters** of the function.

Example:

```
void main ()
{
      int n1, n2, sum;

scanf ("%d%d", &n1, &n2);
sum = add (n1, n2);
printf ("Sum is %d", sum);
}
```

function name

function call

function arguments

In the above example, values of variables n1 and n2 are arguments to the function add(), whereas the variables x and y inside function add() are parameters of the function.