Biyani Girls College
I Internal Examination Solution Sept. 2019
Class: BCA I
Subject- Principles of Programming (BCA- 104)

MM: 40 **Set: B** **Time: 1 ½Hrs**

[I] Very Short Answer Questions (Max 40 words). **(5*2=10)**

1. What do you mean by Literals? Give types of Literals.

Constant values used within a program are known as *Literals*. These constant values occupy memory but do not have any reference like variables. Or as Wikipedia speaks literal is a notation for representing a fixed value within a source code.

There are four types of literals in C programming.

Integer literal
Float or real literal
Character literal
String literal

2. What are Identifiers? Give the rules for Identifier naming.

An identifier is a name that is assigned by the user for a program element such as variable, type, template, class, function or namespace. It is usually limited to letters, digits, and underscores. Certain words, such as "new," "int" and "break," are reserved keywords and cannot be used as identifiers. Identifiers are used to identify a program element in the code.

3. What are Data Types? Give the classification of Data type.

Data types specify how we enter data into our programs and what type of data we enter. C language has some predefined set of data types to handle various kinds of data that we can use in our program. These datatypes have different storage capacities.

C language supports 2 different type of data types:

1. **Primary data types**:

These are fundamental data types in C namely integer(int), floating point(float), character(char) and void.

2. **Derived data types**:

Derived data types are nothing but primary datatypes but a little twisted or grouped together

like **array**, **stucture**, **union** and **pointer**.

3. What are Constants? Give Example.

Constants are like a variable, except that their value never changes during execution once defined.
C Constants is the most fundamental and essential part of the C programming language. Constants in C are the fixed values that are used in a program, and its value remains the same during the entire execution of the program. Constants are also called literals. Constants can be any of the data types.
It is considered best practice to define constants using only upper-case names.

5. What are tokens in C programming?

Tokens are the smallest elements of a program, which are meaningful to the compiler.The following are the types of tokens: Keywords, Identifiers, Constant, Strings, Operators, etc.

[II] Short Answer Questions (Max 80 words). (2*5=10)

1. Write a C program for calculating vote eligibility given the age is entered by the user.

```
2.  #include<stdio.h>
3.
4.  int main()
5.  {
6.     int a ;
7.
8.     //input age
9.     printf("Enter the age of the person: ");
10.    scanf("%d",&a);
11.
12.    //check voting eligibility
13.    if (a>=18)
14.    {
15.                printf("Eigibal for voting");
16.    }
17.    else
18.    {
19.                printf("Not eligibal for voting\n");
20.    }
21.
22.    return 0;
23. }
```

Q. Give the basic structure of a C program along with an example.

Following is the basic structure of a C program.

Documentation	Consists of comments, some description of the program, programmer name and any other useful points that can be referenced later.
Link	Provides instruction to the compiler to link function from the library function.
Definition	Consists of symbolic constants.
Global declaration	Consists of function declaration and global variables.

main( )
{
 	
}
	Every C program must have a main() function which is the starting point of the program execution.

Subprograms	User defined functions.

Example Program :

```
/**
* description: program to find the area of a circle
 *              using the radius r
 */

#include <stdio.h>

#define PI 3.1416

float area(float r);

int main(void)
{
  float r = 10;
  printf("Area: %.2f", area(r));
  return 0;
}

float area(float r) {
   return PI * r * r;
}
```

[III] Long Answer Questions (Max 150 words). **(2*10=20)**

1. Discuss various types of Conditional statements available in C programming.

Conditional statements help you to make a decision based on certain conditions. These conditions are specified by a set of conditional statements having boolean expressions which are evaluated to a boolean value true or false. There are following types of conditional statements in C.

1. If statement
2. If-Else statement
3. Nested If-else statement
4. If-Else If ladder
5. Switch statement

If statement

The single if statement in C language is used to execute the code if a condition is true. It is also called one-way selection statement.

Syntax

```
if(expression)

{

 //code to be executed

}
```

If-else statement

The if-else statement in C language is used to execute the code if condition is true or false. It is also called two-way selection statement.

Syntax

```
if(expression)

{

 //Statements

}

else

{

 //Statements

}
```

Nested If-else statement

The nested if...else statement is used when a program requires more than one test expression. It is also called a multi-way selection statement. When a series of the decision are involved in a statement, we use if else statement in nested form.

Syntax

```
if( expression )

{

 if( expression1 )

 {

 statement-block1;

 }

 else

 {

 statement-block 2;

 }

}

else

{

 statement-block 3;

}
```

If..else If ladder

The if-else-if statement is used to execute one code from multiple conditions. It is also called multipath decision statement. It is a chain of if..else statements in which each if statement is associated with else if statement and last would be an else statement.

Syntax

```
if(condition1)

{

 //statements

}
```

```
else if(condition2)

{

 //statements

}

else if(condition3)

{

 //statements

}

else

{

 //statements

}
```

Switch Statement

switch statement acts as a substitute for a long if-else-if ladder that is used to test a list of cases. A switch statement contains one or more case labels which are tested against the switch expression. When the expression match to a case then the associated statements with that case would be executed.

Syntax

```
Switch (expression)

{

 case value1:

 //Statements

 break;

 case value 2:

 //Statements

 break;

 case value 3:

 //Statements

 case value n:

 //Statements
```

```
break;

Default:

//Statements

}
```

2. Write a C++ program to print the corresponding day according to the number entered by the user i.e. if user enters 1 then it should print Monday.

```c
#include <stdio.h>

int main()
{
    int week;

    /* Input week number from user */
    printf("Enter week number(1-7): ");
    scanf("%d", &week);

    switch(week)
    {
        case 1:
            printf("Monday");
            break;
        case 2:
            printf("Tuesday");
            break;
        case 3:
            printf("Wednesday");
            break;
        case 4:
            printf("Thursday");
            break;
        case 5:
            printf("Friday");
            break;
        case 6:
            printf("Saturday");
            break;
        case 7:
            printf("Sunday");
            break;
        default:
            printf("Invalid input! Please enter week number between 1-7.");
    }

    return 0;
}
```