PPS CT-2

Part-A

1. , → least precedence

2. D = 50 + 12 * 4 / 32 % 4 - 10
   Same precedence = 2 Associativity is from left to right [* / %]

   = 50 + (12 * 4) / 32 % 4 - 10
   = 50 + (48/32) % 4 - 10
   = 50 + (1 % 4) - 10
   = 50 + 1 - 10
   = 41

∴ Order of Evaluation:-
*/%+-=

3. int $ main; // Invalid since there is no space bet^n data type & variable-name & (identifier)

4. The format specifier that is used to read or write a character is %c

5. Result of logical or relational expression in C is 0 or 1

6. Relational

7. The sizeof() operator is a unary type of operator used to calculate size of the data types (in bytes)

8. Result = 2, 4

9. Indentation is used to write the pseudo code with hierarchy

10. A binding is dynamic if it first occurs during execution or can change during execution of program

Part B:-

11. Pseudocode SUM_AVG

```
BEGIN
READ A, B, C
CALCULATE sum = A+B+C
CALCULATE average = sum/3
PRINT sum and average
END
```

12.
```
#include <stdio.h>
void main(){
    int P, T;
    float R, SI;
    P = 1000;
    T = 3;
    R = 8.5;
    SI = P*R*T/100;
    printf("Simple Interest = %f", SI);
```

13. = is assignment operator. It is used to assign value to variable.

== is relational operator. It is used to compare two operands and check if they are equal to each other

Eg:- i=5 :- means "5 is being assigned to variable i or i now contains value of 5

i==5 :- it will compare value of i and 5. If both are equal, returns true (1) else, returns false (0)

14. • Global variables :-

Storage :- They are stored wherever the linker puts them. Usually stored in data segment of the memory

Lifetime :- They exist only for duration of the program and are initialized to zero (if not explicitly initialized)

• Static variables :-

Storage :- They are stored wherever the linker puts them. Usually stored in data segment of the memory

Lifetime :- They retain their value during function calls and can exist for the lifetime of program.

• Local variables / Register variables :-

Storage :- They are stored in stack.

Lifetime :- Their lifetime is limited to the scope of the function in which they are defined. Once this function exits, they are not accessible.

• Free Memory (Dynamic Memory) :-

Storage :- They are stored in heap

Lifetime :- This memory remains allocated unless it is explicitly freed using free()
~~allocating~~

• C Program Instructions :-

Storage :- The actual compiled instructions of program are stored in text/code segment

Lifetime :- These instructions are loaded into memory when the program runs and exist for the duration of program execution.

15.
```
#include <stdio.h>
int main(){
    printf("Size of int = %ld\n", sizeof(int));
    printf("Size of float = %ld\n", sizeof(float));
    printf("Size of char = %ld\n", sizeof(char));
    printf("Size of double = %ld\n", sizeof(double));
    return 0;
}
```

18. Part C :-

16. Algorithm :-

START
DECLARE name, bp, hra, da, pf, gs
READ name, bp, hra, da from the user
Calculate gs = bp + hra + da + pf → (12/100)*bp
DISPLAY gs
STOP

Program :-
```
#include <stdio.h>
int main(){
    char name[50];
    gets(name); // Input name from user
    .
    .
}
```

17. Flowchart :-