

P and PS (Theory)

Assignment No. 1

1) What is the difference between = symbol and == symbol?

Ans - = and == both are operators in C language.

→ = is assignment operator used to assign values. It's Associativity is Right to left.

→ == is Relational operator which is used to compare two values.

eg: `int a = 5, b = 10, c;`

`c = a;` // Here value of a is assigned to c.

`if (a == b)` // Here we compare value of a and b.

2) When a "switch" statement preferable over an "if" statement?

Ans - • Switch statement works better than multiple if statements when we are giving input directly without any condition checking in the statements.

• Switch works well when we want to increase the readability of the code and many alternative available.

• When the selection condition lies between range instead of calculating the range make them as cases with numbers assigned to it properly and use switch case.

3) In a switch statement, what will happen if a break statement is omitted.

Ans- If, in a switch statement we omit the break statement then it will execute all cases from the case started.
eg:

```
switch (ch) {
    case 1:
    case 2:
    case 3:
        printf("We are in case 3");
    case 4:
        printf("We are in case 4");
    default:
        printf("We are in default case");
}
```

If $ch = 3$, then output will be

$$\left\{ \begin{array}{l} \text{we are in case 3} \\ \text{we are in case 4} \\ \text{we are in default case.} \end{array} \right\}$$

4) Is that possible to store 32768 in an int data type var?

Ans- In DOS based 16-bit architecture, it is not possible to store 32768 in an int datatype variable because int is 2 byte size in DOS based architecture and ranges from -32768 to 32767.

5) What are the valid places where the programs can apply continue statement?

Ans- The continue statement is used inside loops when a continue statement is encountered inside a loop, control jumps to the beginning of the loop for the next iteration, skipping the execution of statements inside the body of loop for the current iteration.

eg:-

```
#include <stdio.h>
int main () {
    int i;
    for (i = 1; i <= 5; i++) {
        if (i == 3)
            continue;
        printf ("%d", i);
    }
    return 0;
}
```

Output :

1 2 4 5

// here 3 is skipped from iteration.

6) What are the advantages of an array over a simple variable?

Ans- • If we require a bunch of variables for the same data type then simply declare an array rather than declaring variables.
eg: int a, b, c, d, e; // declaring 5 int variables individually.
int a[5]; // declaring 5 int type variables by arrays.

- If we declare an array dynamically then we save memory.
- By array we can declare elements in multi dimensional format where individual variable declaration doesn't facilitate it.
- By operation on array rather than individual variables reduce length of code and easily readable.

7) Describe how array can be passed to a user defined function?

Ans - If we create a user defined function and pass an array to them then actually we don't pass the whole array we can just pass the base address of array.
eg: // sum of 5 numbers.

```
#include <stdio.h>
int sum(int a[], int);
int main()
```

```
{
    int a[5], i, s;
    printf("Enter 5 elements in array: ");
    for (i = 0; i < 5; i++)
    {
```

```
        scanf("%d", &a[i]);
    }
```

```
    s = sum(a, 5); // Passing base address of array
    printf("Sum = %d", s);
    return 0;
}
```

```
int sum (int a[], int n)
{
    int i, s=0;
    for (i=0; i<n; i++)
        s += a[i];
}

return s;
}
```

8) Explain the difference between delete and delete [].

Ans- Delete is an operator which is used for releasing the dynamically allocated memory in C++.

→ delete operator deallocates memory and calls the destructor for a single object created with new operator.

→ delete [] operator deallocates memory and calls destructor for an array of objects created with new operator.

9) What do you mean by nested for loop in C programming?

Ans- If in a C program a for loop is inside a for loop exist, then this is called a nested for loop.

eg:

```
*
* *
* * *
* * * *
* * * * *
```

```
#include <stdio.h>
int main() {
    int i, j;
    for (i = 0; i < 5; i++) {
        for (j = 0; j <= i; j++)
            printf("x ");
        printf("\n");
    }
    return 0;
}
```

10> What is the difference between while and do-while loop in C?

Ans -

while	Do-while
<ul style="list-style-type: none"> • Condition is checked first then statement is executed. • No semicolon at the end of while condition. • while is entry controlled. • If there is a single statement brackets are not required. • while (condition) { statement(s); } 	<ul style="list-style-type: none"> • Statement is executed at least once then condition is checked. • Semicolon at the end of while condition. • Do while is exit controlled. • Brackets are always required. • do { statement(s); } while (condition);