

NAME : M. ARJUN

UNIQUE ID : E0222054

SUBJECT : PROBLEM SOLVING - C

DATE : 30/09/2022

1) Looping Statements are introduced in a program to reduce the code complexity and when we want to execute the same statement again and again.

There are 3 Types of Loops:-

### ① For Loop :-

First of all, to use for loop we need to know the limit and when the limit is satisfied the for loop breaks and executes the rest of the program.

#### Sample Program:-

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    int a;
```

```
    for (a = 0 ; a < 10 ; a++)
```

```
{
```

```
    printf("Printing Values from 0 to 9\n");
```

```
    printf("%d\n", a);
```

```
}
```

```
    return 0;
```

```
}
```



## OUTPUT :-

Printing Values from 0 to 9

0

Printing Values from 0 to 9

1

...

...

Printing Values from 0 to 9

9

## ② While loop:-

To Use while loop we don't need to know the end value or limiting value. Once the condition is satisfied the while loop breaks.

### Sample Program:-

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i = 1, n;
```

```
    printf("Enter the Limit : ");
```

```
    scanf("%d", &n);
```

```
    while (i <= n)
```

```
{  
    printf("Enter the Limit :  
}
```

```
{  
    printf("%d", i);  
    i++;  
}  
return 0;
```

```
}
```

OUTPUT :-

Enter the Limit : 5

1  
2  
3  
4  
5

### ③ Do While Loop:-

This Loop doesn't check the condition while executing first time and this loop is also known as exit check loop. This process continues until the condition is false.

### Sample Program:-

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    int i = 0, n;
```

```
    printf ("\n Enter The Limit : ");
```

```
    scanf ("%d", &n);
```

```
    do
```

```
    {
```

```
        printf ("\n %d", i);
```

```
        i += 2;
```

```
    } while (i <= n);
```

```
    return 0;
```

```
}
```

### OUTPUT:-

Enter The Limit : 8

0

2

4

6

8



2. Write about Conditional Statements with an example.

Conditional Statement is used when we have some conditions, if that condition is true the code inside that if block executes.

There are 3 Conditional Statements.

### ① If

If statement is a simple decision-making and branching statement and it is used to control the flow of the program execution. The if statement is written with the if keyword, followed by a condition in parentheses, with the code to be executed between curly brackets.

### Sample Program:-

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    int age;
```

```
    printf("\n Enter Age : ");
```

```
    scanf("%d", &age);
```

```
    if (age >= 18)
```

```
{
```

```
        printf("\n %d is eligible for vote", age);
```

```
}
```

return 0;

}

OUTPUT:

Enter Age : 15

## ② If Else :-

The else block is optional, It is used only when the condition in the if block is false.

### Sample Program :

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int age;
```

```
    printf("Enter Age: ");
```

```
    scanf("%d", &age)
```

```
    if (age >= 18)
```

```
    {
```

```
        printf("Eligible for Vote.")
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("Not Eligible")
```

```
    }
```

```
    return 0;
```

```
}
```

## OUTPUT:-

Enter Age: 22

Eligible For Vote.

### ③ Else if

Else if is used to specify and customize the program more effectively. If the 'if' condition is false then it comes to ~~if~~ elseif condition.

### Sample Program:-

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    int a;
```

```
    printf("Enter any value : ");
```

```
    scanf("%d", &a);
```

```
    if (a < 0)
```

```
    { printf("Negative Value);
```

```
    }
```

```
    else if (a == 0)
```

```
    { printf("Value is 0);
```

```
    }
```



```
else  
{ printf("Positive Value") }  
return 0;  
}
```

Output :-

Enter any value : 0

Value is 0.

③ Write about arrays (int, float, char) with an example

An array is a variable that can store multiple values. For example if you want to store 10 integers, ~~you~~ first <sup>you</sup> need to define 10 variables. Array can only store homogenous datatypes values.

① Int Array:-

Sample Program:-

```
#include <stdio.h>
```

```
int main()
```

```
{  
    int int i, a[15], n;  
    printf("\n Enter the Limit:");  
    scanf("%d", &n);
```

```
    for (i = 0; i < n; i++)
```

```
    {  
        printf("\n Enter Value :");  
        scanf("%d", &a[i]);
```

```
    }
```

```
    for (i = 0; i < n; i++)
```

```
    {  
        printf("\n %d", a[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

OUTPUT :

Enter the Limit : 3

Enter Value : 1

Enter Value : 2

Enter Value : 3

1

2

3

② Float Array :

~~1.1~~

Sample Program

```
#include <stdio.h>
```

```
int main()
```

```
{ int i, n;
```

```
float 1.1 a[100]; 1.1
```

```
printf("\n Enter Limit : ");
```

```
scanf("%d", &n);
```

```
for(i = 0; i < n; i++)
```

```
{ printf("\n Enter Value : ");
```

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

```
}
```



```

for (i = 0; i < n; i++)
{
    printf (" \n %.0.2f ", a[i]);
}
return 0;
}

```

OUTPUT:

Enter Limit : 2

Enter Value : 1

Enter Value : 5.3685

1.00

5.36

③ Char Array :-

Sample Program :

```
#include <stdio.h>
```

```
int main()
```

```
{
    char C[5] = {'A', 'R', 'J', 'U', 'N'}
    int i;

```

```
for (i = 0; i < 5; i++)
```

```
{  
    printf("The word [%d] is %c\n", i+1, word[i]);
```

```
}  
    return 0;
```

```
}
```

OUTPUT:

The word [1] is A

The word [2] is R

The word [3] is J

The word [4] is U

The word [5] is N