

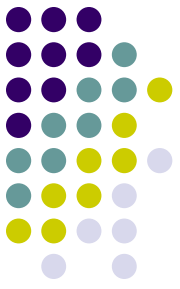
# Paper 102: Programming & Problem solving through C

---

## Lecture-06:Unit-I C Fundamentals



# for loop

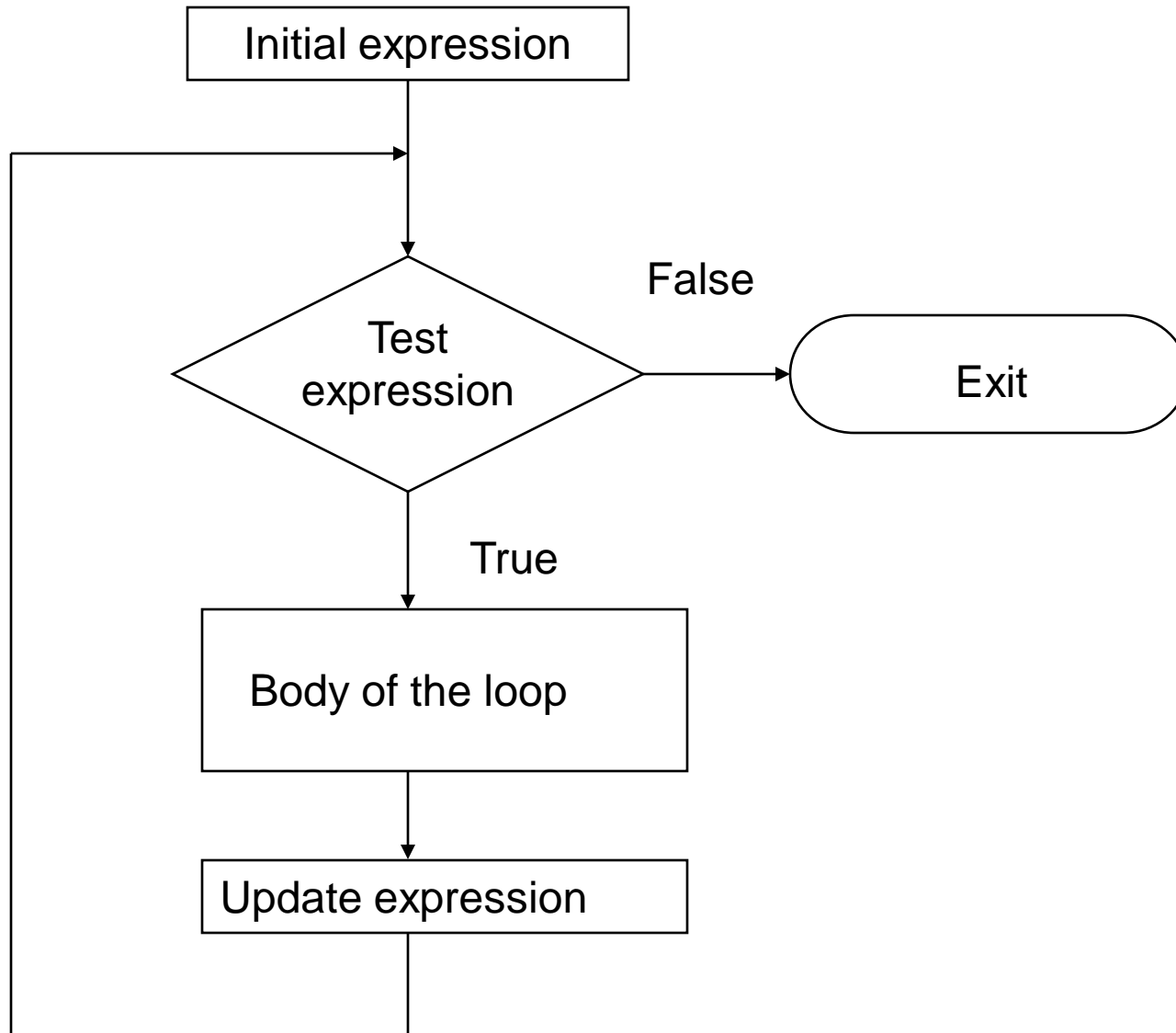


This loop is useful when a block of statement is to be executed a number of times.

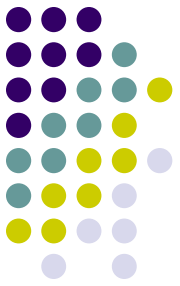
```
for (initial expression; test expression; update expression)
    statement/block of statement
```

1. The initial expression is executed only once
2. The test expression is tested once **before** every iteration. If it is true the block of statement is executed once else the loop terminates.
3. The update expression is executed once **after** every iteration.

# Flowchart of the for loop statement



# for loop:Example-1



```
void main()
{
    int i,n,square,sum=0;
    n=5;
    for(i=1;i<=n;i++)
    {
        square=i*i;
        printf("The square of %d is %d \n",i, square);
        sum=sum + square;
    }
    printf("The sum of all the squares from 1 to %d is = %d", n,sum);
}
```

# for loop:Example-2



The for loop can also be nested

```
void main()
{
    int j,k;
    for(j=1;j<=3;j++)
    {
        for(k=1;k<=j;k++)
        {
            printf("\n j=%d k=%d",j,k);
        }
    }
}
```

# Valid for loops



```
for(;j<=n;j++)
```

```
{
```

```
Statements
```

```
}
```

```
for(;;k++)
```

```
{
```

```
Statements
```

```
}
```

```
for(j=1,k=1;j<=4;j++,k++)
```

```
{
```

```
statements
```

```
}
```

# The break and continue statements



- The **break statement** terminates the execution of a loop and the control is transferred to the statement immediately following the loop
  - Can be written as **break;**
- The **continue statement** is used to bypass the remaining statement after it and the control is transferred to the beginning of the loop.
  - Can be written as **continue;**

# The break statement: Example



```
void main()
{
    int a=1,b=1;
    while(a<=10)
    {
        while(b<=10)
        {
            if (a==b)
                break;
            else
                printf("a=%d b=%d",a,b);
            b++;
        }
        a++;
    }
}
```



# The continue statement: Example



```
void main()
{
    int a,b;
    for(a=1;a<=2;a++)
    {
        for(b=1;b<=2;b++)
        {
            if (a==b)
                continue;
            printf("a=%d b=%d",a,b);
        }
    }
}
```

# switch-case statement



This control statement allows decision to be made from a number of choices.

```
switch (variable name)
{
    case constant 1:
        statements
    case constant 2:
        statements
    .
    .
    .
    case constant n:
        statements
    default:
        statements
}
```

1. It is useful if a variable is compared with different constants, and in case it is equal to a constant.
2. The constant in the case statement can be of char or int data type only.

# switch-case statement:Example-1



```
void main()
{  int i=2;
   switch (i)
   {
       case 1:
           printf("This is case constant 1 \n");
       case 2:
           printf("This is case constant 2 \n");
       case 3:
           printf("This is case constant 3 \n");
       default:
           printf("This is a default statement \n");
   }
}
```

# switch-case statement:Example-2



```
void main()
{  char c='a';
   switch (c)
   {
       case 'a':
           printf("This is case constant a \n");
       case 'e':
           printf("This is case constant e \n");
       case 'o':
           printf("This is case constant o \n");
       default:
           printf("This is a default statement \n");
   }
}
```

# switch-case statement:Example-3



```
void main()
{ char c='E';
  switch (c)
  {
    case 'a':
    case 'A':
        printf("This is an alphabet A \n");
    case 'e':
    case 'E':
        printf("This is an alphabet e \n");
    case 'o':
        printf("This is an alphabet o \n");
    default:
        printf("This is a default statement \n");
  }
}
```

# switch-case statement:Example-4



```
void main()
{  char c='E';
   switch (c)
   {
       case 'a':
       case 'A':
           printf("This is an alphabet A \n");
           break;
       case 'e':
       case 'E':
           printf("This is an alphabet E \n");
           break;
       case 'o':
           printf("This is an alphabet o \n");
           break;
       default:
           printf("This is a default statement \n");
   }
}
```



# Class Assignment

- Write a program to input the rollno., marks in 3 subjects and marks in one additional subject. Compute the total marks, excess additional marks should be added to the total marks and determine the division. No division should be given if fail in 1 or more subjects.
- Write a program to generate Fibonacci series and print their sum and average.
- Write a program to generate all the prime numbers till N.
- Write a program to compute the series below:  
 $1 - 1/2^2 + 1/3^2 - 1/4^2 + \dots$