Control Structures and Loops

PRESENTED BY: SATYA PRAKASH PATEL
EMAIL: SATYAPATEL.IND@GMAIL.COM

Ternary operator ?:

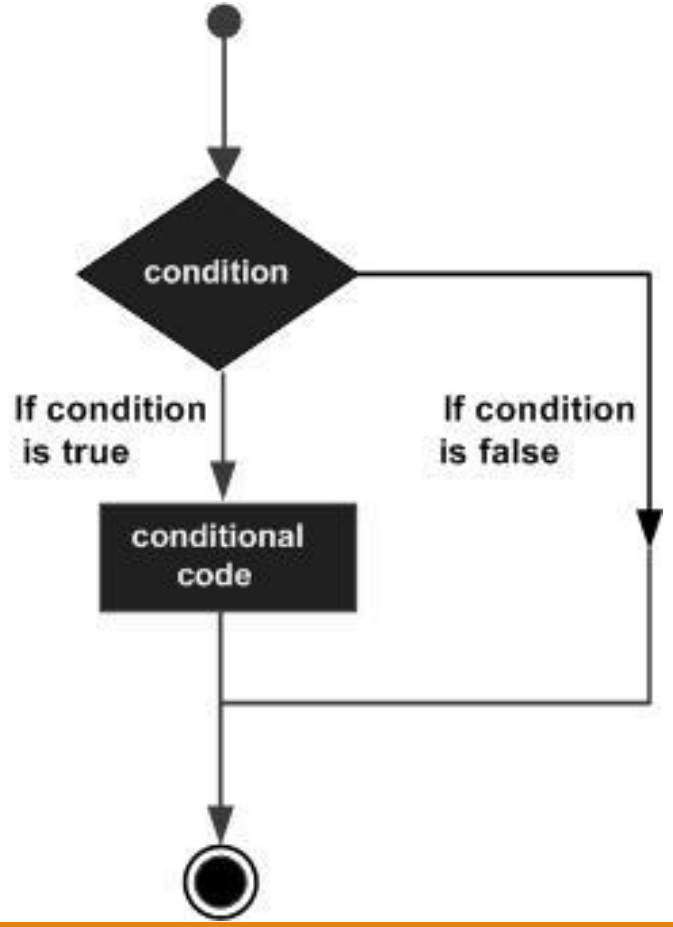
Syntax: <Condtion>?<True Block>:<False Block>
Example: 2>3? printf(“2”): printf(“3”);

Flow chart

The ternary operator is right-associative. The expression a ? b : c ? d : e is evaluated as a ? b : (c ? d : e), not as (a ? b : c) ? d : e

Conditional Structures

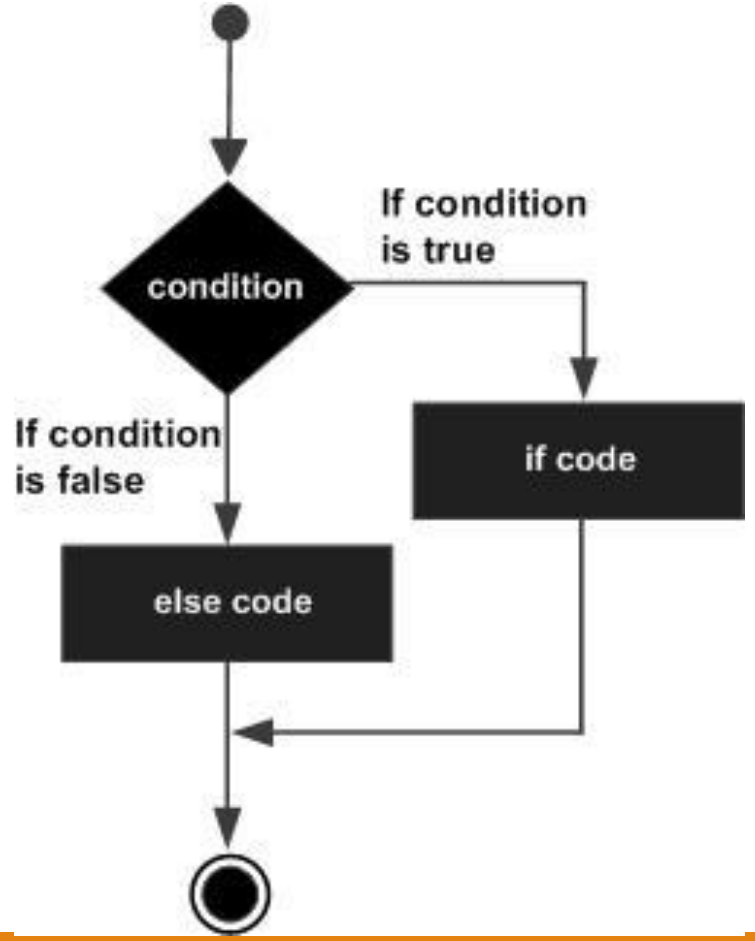
```
if(boolean_expression) {
   /* statement(s) will execute if the boolean expression is true */
}
```



```
if (testExpression)
{
  // statement(s)
}
```

```
if (number < 0)
    {
        printf("You entered %d.\n",
number);
    }
```

```
if(boolean_expression) {
   /* statement(s) will execute if the boolean expression is true */
} else {
   /* statement(s) will execute if the boolean expression is false */
}
```



```
if( number%2 == 0 )
        printf("%d is an even integer.",number);
    else
        printf("%d is an odd integer.",number);
```

if...else Ladder (if...else if....else Statement)

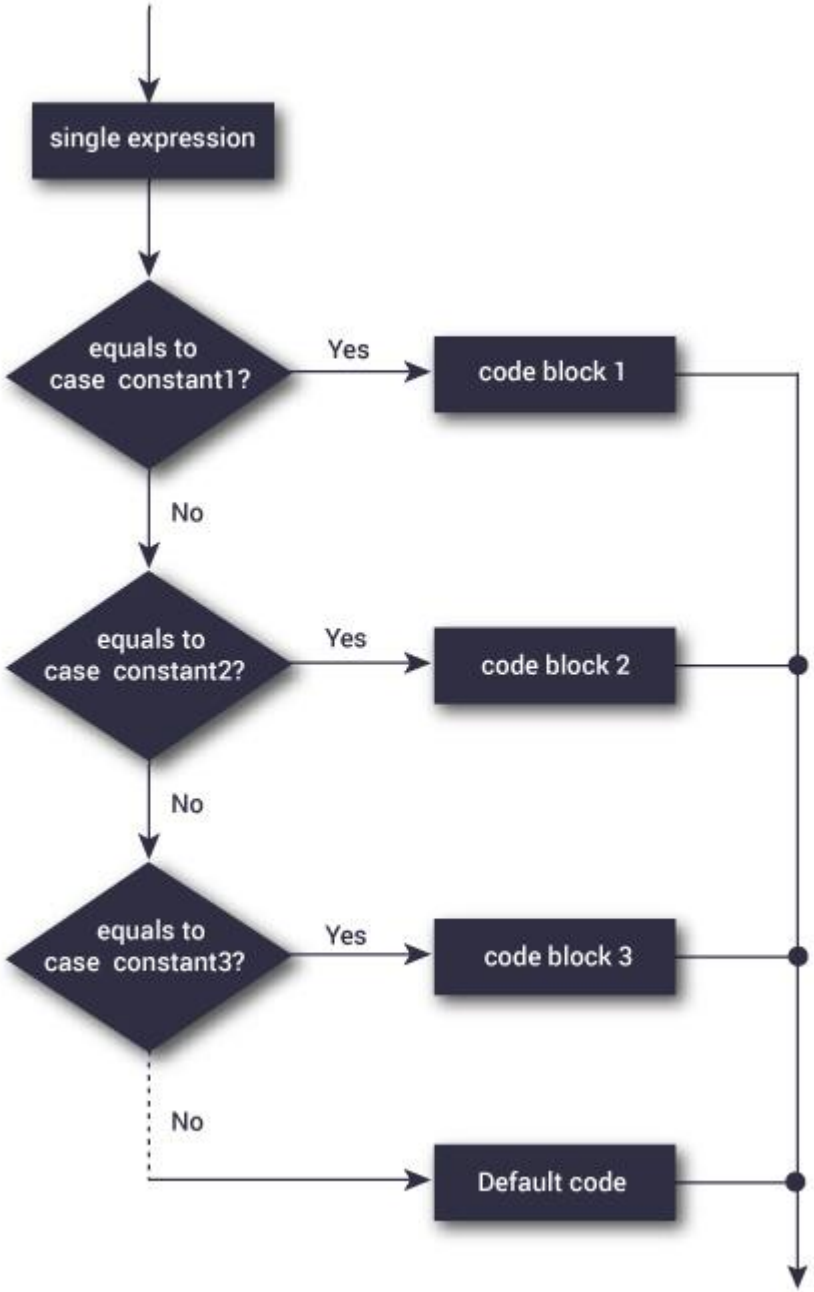
```
if (testExpression1) {
   // statement(s)
}

else if(testExpression2)

{   // statement(s)}

else if (testExpression 3)

{   // statement(s)

}

else

{   // statement(s)

}
```

Nested if…else

```
if (number1 >= number2)
 {
    if (number1 == number2)
    {       printf("Result: %d = %d",number1,number2);}
    else {  printf("Result: %d > %d", number1, number2);       }
 }
else
  {
     printf("Result: %d < %d",number1, number2);
  }
```

Switch case

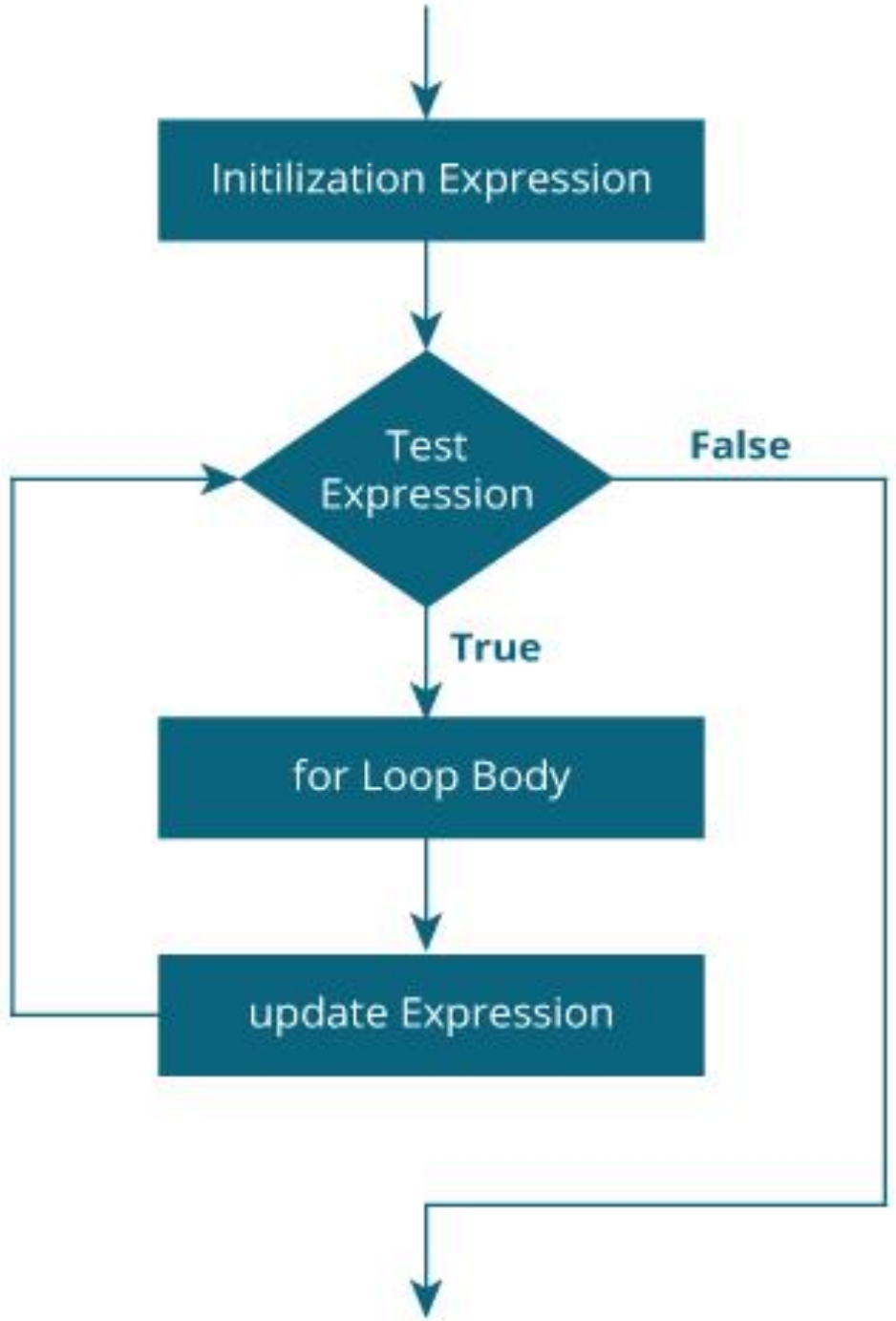
```
switch (n)
{    case constant1:
       // code to be executed if n is equal to constant1;
       break;
    case constant2:
       // code to be executed if n is equal to constant2;
       break;
    default:
       // code to be executed if n doesn't match any constant
}
```


single expression
equals to case constant1?
Yes
code block 1
No
equals to case constant2?
Yes
code block 2
No
equals to case constant3?
Yes
code block 3
No
Default code

for Loop

```
for (initializationStatement; testExpression; updateStatement)
{
    // codes
}
```

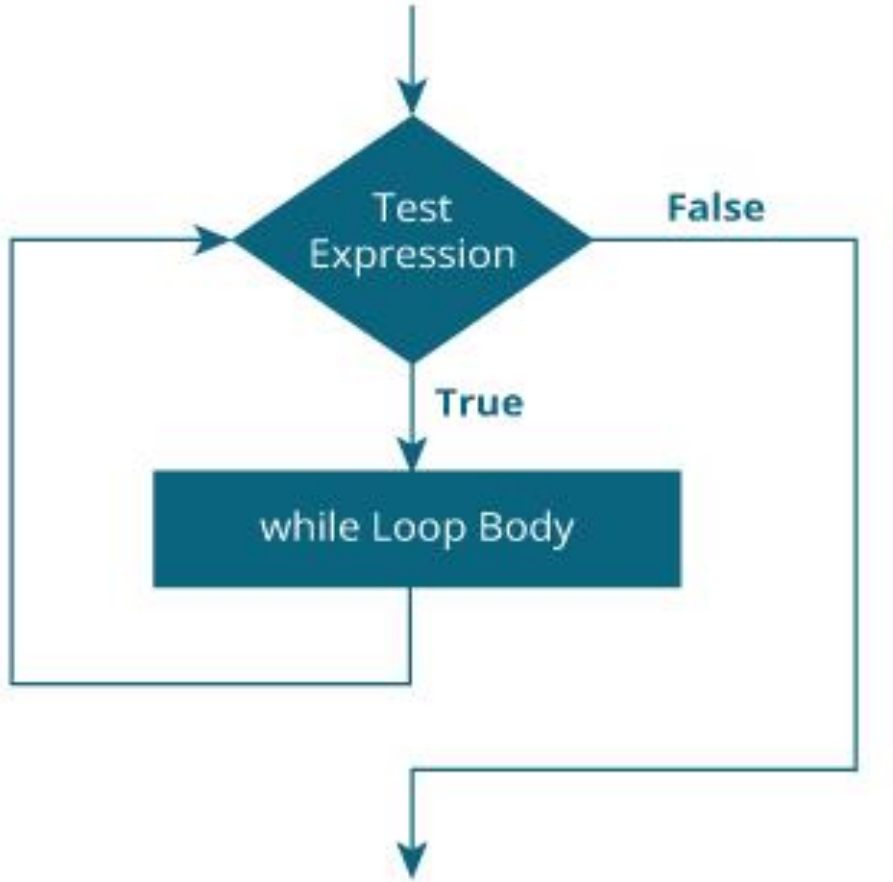
Flow chart for for loop



While loop

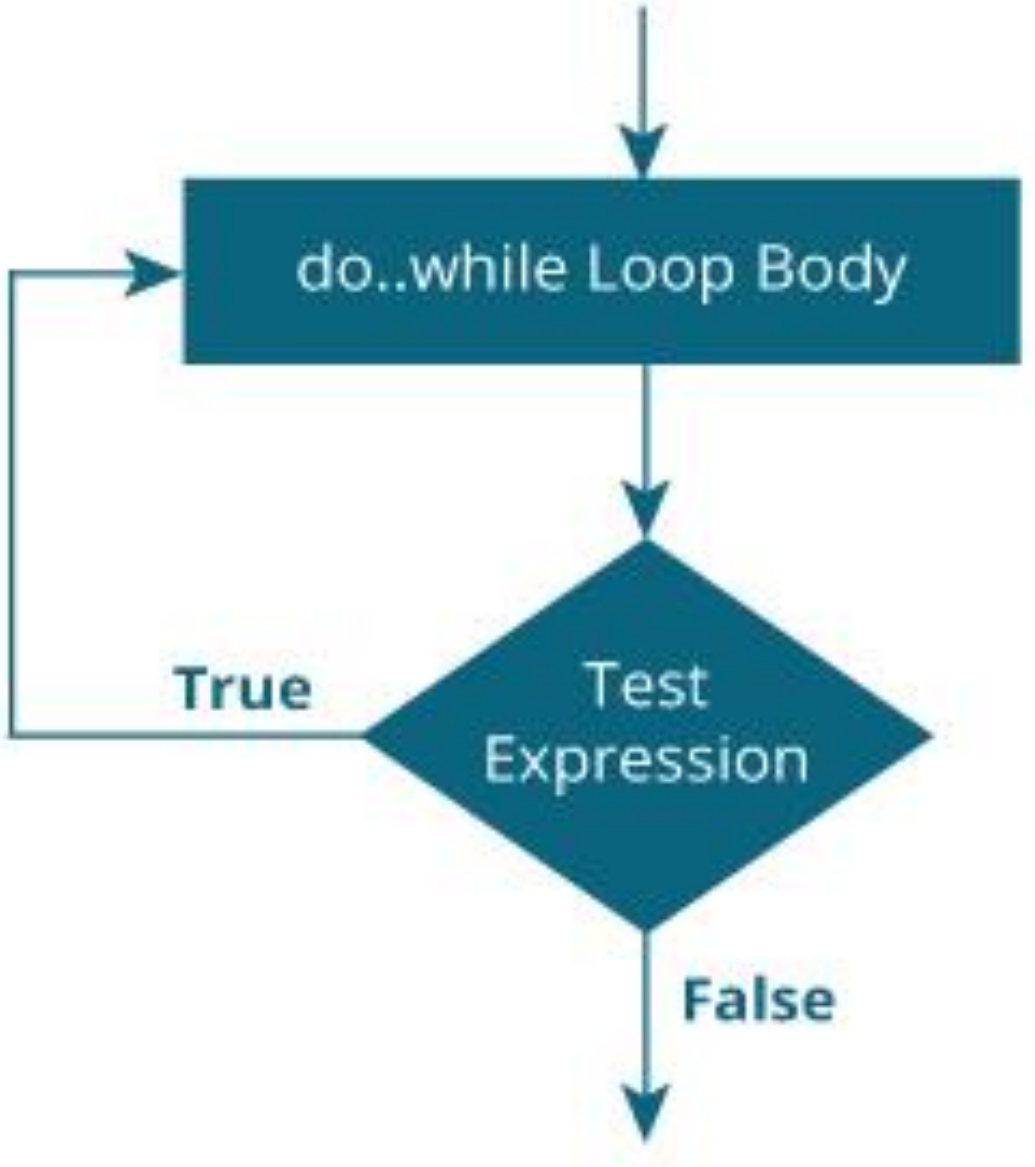
WHILE

```
while (testExpression)
{
   //codes
}
```



do...while loop Syntax

```
do
{
  // codes
}
while (testExpression);
```



break Statement

The break statement terminates the loop (for, while and do...while loop) immediately when it is encountered.

Its syntax is:

```
break;
```

```
while (testExpression) {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
```

```
do {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
while (testExpression);
```

```
for (init; testExpression; update) {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
```

Source: https://www.programiz.com/

continue Statement

The continue statement skips statements after it inside the loop.

Its syntax is:

```
continue;
```

```
while (testExpression) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

```
do {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
} while (testExpression);
```

```
for (init; testExpression; update) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

C goto Statement

used to alter the normal sequence of a C program.

Syntax of goto statement:

```
goto label;
... .. ...
... .. ...
... .. ...
label:
statement;
```

Reasons to avoid goto statement

The use of goto statement may lead to code that is buggy and hard to follow.

```
one:
for (i = 0; i < number; ++i)
{
    test += i;
    goto two;
}
two:
if (test > 5) {
  goto three;
}
... .. ...
```

Question 1

What will be the output of following program ?

```
#include <stdio.h>
void main()
{
        const char var='A';
        ++var;
        printf("%c",var);
}
```

OPTIONS

A. B
B. A
C. 66
D. ERROR

Question 2

WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM ?

```
#include <stdio.h>

void main()

{

        int x=(20 || 40 ) && (10);

        printf("x= %d",x);

}
```

OPTIONS

A. x= 1

B. x= 0

C. x= 60

D. x= 70

Question 3

WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM ?

```
int main()
{
  int x, y = 5, z = 5;
  x = y == z;
  printf("%d", x);
  return 0;
}
```

OPTIONS

A. 0

B. 1

C. 5

D. Compiler Error

Question 4

WHICH OF THE FOLLOWING OPERATORS HAS AN ASSOCIATIVITY FROM RIGHT TO LEFT?

OPTIONS

A. +=

B. <<

C. ==

D. <=

Question 5

WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM ?

```
void main()
{       int x;
        x= (printf("AA")||printf("BB"));
        printf("%d",x);
        printf("\n");
        x= (printf("AA")&&printf("BB"));
        printf("%d",x);
}
```

OPTIONS

A. AABB1
   AABB1

B. 1
   1

C. AA1
   AABB1

D. AABB1
   AA1

Home work1 : analyse the question

```
#include<stdio.h>
void main()
{       int x,y;
        x= printf("%d\n",7);
         y= printf("\n%d BIT",7);
        printf("\nx= %d, \ny= %d ",x, y);

}
```

Home work 2: analyse the question

```
#include<stdio.h>
void main()
{       int x,y;
    x= scanf("%d %d %d",&x, &y, &x);
    printf("\nx= %d,",x);
    y= scanf("%d %d %d",&x, &y, &x);
        printf("\nx= %d, \ny= %d ",x, y);
}
```

Home work 3: analyse the question

```
#include<stdio.h>

void main()

{       int x,y,z;
  ◦ scanf(“%d %d %d”, &x,&y,&z)

   x>z?x>y?printf("x"):printf("y"):printf("z");

    printf("\n\n\n\n\nx=%d\ny=%d, \n z=%d", x,y,z);

}
```

Input case

A. X=1,y=2,z=3
B. X=1,y=3,z=2
C. X=3,y=1,z=2
D. X=3,y=2,z=1

Queries and Feedback