Unit - III   CHAPTER - 10

Flow of Control

PART – I

Choose the correct answer.

1. What is the alternate name of null statement?

    (A) No statement   **(B) Empty statement** (C) Void statement   (D) Zero statement

2. In C++, the group of statements should enclosed within:

    (A) { }   (B) [ ]   (C) ( )   (D) < >

3. The set of statements that are executed again and again in iteration is called as:

    (A) condition   **(B) loop**   (C) statement   (D) body of loop

4. The multi way branching statement:

    (A) if   (B) if … else   **(C) switch**   (D) for

5. How many types of iteration statements?

    (A) 2   **(B) 3**   (C) 4   (D) 5

6. How many times the following loop will execute? for (int i=0; i<10; i++)

    (A) 0   **(B) 10**   (C) 9   (D) 11

7. Which of the following is the exit control loop?

    (A) for   (B) while   **(C) do…while**   (D) if…else

8. Identify the odd one from the keywords of jump statements:

    (A) break   **(B) switch**   (C) goto   (D) continue

9. Which of the following is the exit control loop?

    (A) do-while   (B) for   (C) while   (D) if-else

10. A loop that contains another loop inside its body:

    (A) Nested loop   (B) Inner loop   (C) Inline loop   (D) Nesting of loop

Part – II

Answers to all the questions (2 Marks):

1. What is a null statement and compound statement?

➤ The "null or empty statement" is a statement containing only a semicolon (;)

➤ C++ allows a group of statements enclosed by pair of braces {}. This group of statements is called as a compound statement or a block.

2. What is selection statement? Write it's types?

➤ The selection statement means the statement (s) are executed depends upon a condition. If a condition is true, a true block (a set of statements) is executed otherwise a false block is executed. This statement is also called decision statement or selection statement

**Types**: If, if else, Nest if, if -else-if ladder, The ?: Alternative to if- else, Switch statement

3. Correct the following code sigment:

```
if (x=1)
p = 100;
else
p = 10;
```

```
if (x= =1)
p = 100;
else
p = 10;
```

4. What will be the output of the following code:

```
int year;
cin >> year;
if (year % 100 == 0)
if ( year % 400 == 0)
cout << "Leap";
else
cout << "Not Leap year";
```

If the input given is

(i)	2000	-	**Leap**
(ii)	2003	-	**Not Leap year**
(iii)	2010	-	**Not Leap year**

5. What is the output of the following code?

```
for (int i=2; i<=10 ; i+=2)
cout << "\n" <<i;
```

```
2
4
6
8
10
```

6. Write a for loop that displays the number from 21 to 30.

```
#include <iostream>
using namespace std;
int main()
{
clrscr();
int i;
for (i=21; i<=30 ; i++)
cout << "\n" <<i;
getch();
}
```

7. Write a while loop that displays numbers 2, 4, 6, 8.......20.

```
#include <iostream>
using namespace std;
int main()
{
clrscr();
int i;
while (i=2; i<=20 ; i+=2)
cout << "\n" <<i;
getch();
}
```

8. Compare an if and a ? : operator.

if	? : Operator
if the condition is true then a true-block executed, otherwise the true-block is skipped	The conditional operator (or) Ternary operator is an alternative for 'if else statement'.

Syntax:
 	
    if (expression)
    true-block;
    statement-x;
	Syntax:
	
	    expression 1? expression 2 : expression 3

Part – III

Answers to all the questions (3 Marks):

1. Convert the following if-else to a single conditional statement:

if (x >= 10)
a = m + 5;
else
a = m;

if (x >= 10 ? a = m + 5 : a = m)

2. Rewrite the following code so that it is functional:

v = 5;
do;
{
total += v;
cout << total;
while v <= 10

```
int v = 5;
do
{
cout << v;
total += v;
}
while (v <= 10);
```

3. Write a C++ program to print multiplication table of a given number.

```cpp
#include<iostream>
using namespace std;
int main()
{
int num;
cout<<"Enter Number To Find Multiplication table ";
cin>>num;
for(int a=1;a<=10;a++)
{
 cout<<num<<" * "<<a<<" = "<<num*a<<endl;
}
return 0;
}
```

Output:

```
Enter Number To Find Its Multiplication table  3
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30
Press any key to continue
```

4. Write the syntax and purpose of switch statement.

syntax of switch   *purpose of switch statement*

```
switch(expression)
{
case constant 1:
statement(s);
break;
case constant 2:
statement(s);
break;
.
.
default:
statement(s);
}
```

➢ The switch statement is a multi-way branch statement.

➢ A switch statement can only work for quality of comparisons.

➢ No two case labels in the same switch can have identical values.

5. Write a short program to print following series:

(a) 1 4 7 10...... 40

```
#include<iostream>
using namespace std;
int main()
{
 clrscr();
 int i,a=-1,b;
 for(i=1;i<=40;i+=3)
 {
     a*=-1;
     b=i;
     b*=a;
     cout<<b<<"   ";
 }
 getch();
}
```

Part – IV

Answers to all the questions (5 Marks):

1. Explain control statement with suitable example.

➢ Control statements are statements that alter the sequence of flow of instructions.

In a program, statements may be executed sequentially, selectively or iteratively. Every programming languages provides statements to support sequence, selection (branching) and iteration.

If the Statements are executed sequentially, the flow is called as sequential flow.

In some situations, if the statements alter the flow of execution like branching, iteration, jumping and function calls, this flow is called as control flow.

The **sequential statement** are the statements, that are executed one after another only once from top to bottom. These statements do not alter the flow of execution. These statements are called as sequential flow statements. They are always end with a semicolon (;).
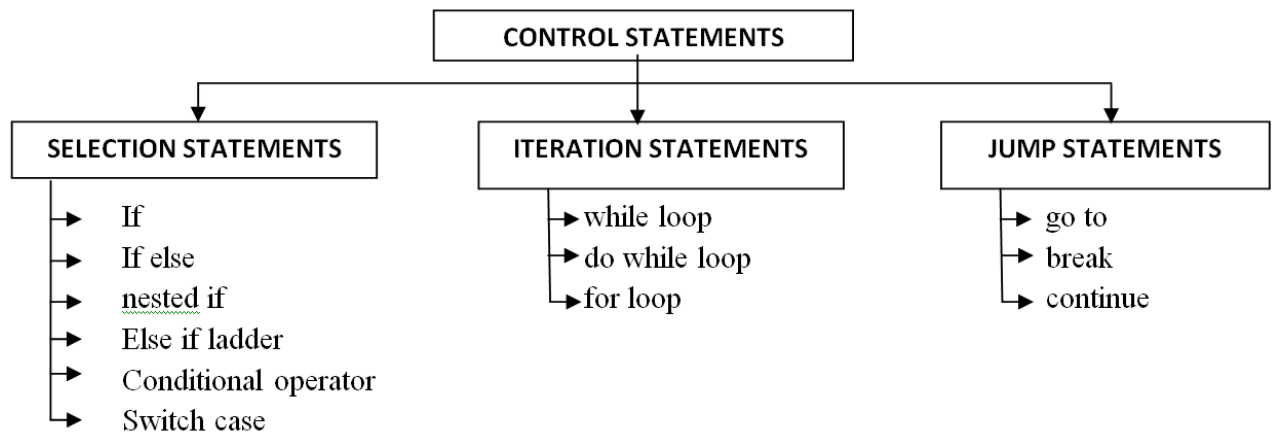
1) **The selection statement** means the statement (s) are executed depends upon a condition. If a condition is true, a true block (a set of statements) is executed otherwise a false block is executed. This statement is also called **decision statement** or **selection statement.**

2) **Iteration Statement (looping)** is use to execute a set of statements repeatedly until a condition is satisfied.
➢ If a condition evaluates to true, the set of statements (true block) is executed again and again.
➢ This is also known as **looping statement** or iteration statement.

3) **Jump Statement** are used

➢ (1) *goto*
➢ (2) *break*
➢ (3) *continue*

2. What entry control loop? Explain any one of the entry control loop with suitable example.
    Loop body will be executed first, and then condition is checked. ... If Test condition is false, loop body will be executed once.
Examples of Entry Controlled Loop:
    ✓ *for loop*
    ✓ *while loop*
for loop:
    The for-loop is the easiest looping statement which allows code to be executed repeatedly.
The general syntax is: **Flow chart:**

-+
```
for (initialization(s); test-expression; update expression(s))
{
Statement 1;
Statement 2;
………….
}
    Statement-x;
```
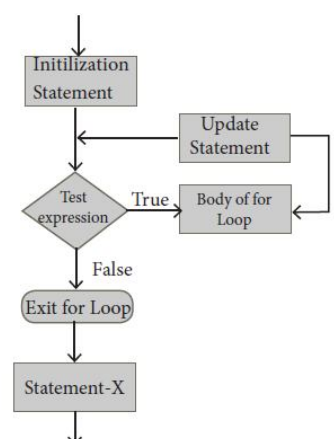

Example Program:

```
#include <iostream>
using namespace std;
int main ()
{
int i;
for(i = 0; i< 5; i ++ )
cout<< "value of i : " <<i<<endl;
return 0;
}
```

Output
    value of i : 0
    value of i : 1
    value of i : 2
    value of i : 3
    value of i : 4

3. Write a program to find the LCM and GDC of two numbers.

To find the LCM of two numbers

```
#include <iostream>
using namespace std;
int main()
{
    int n1, n2, max;
    cout << "Enter two numbers: ";
    cin >> n1 >> n2;
    max = (n1 > n2) ? n1 : n2;
    do
    {
        if (max % n1 == 0 && max % n2 == 0)
        {
            cout << "LCM = " << max;
            break;
        }
        else
            ++max;
    } while (true);
        return 0;
}
```

Output:

```
Enter two numbers: 12
18
LCM = 36
```

To find the GDC of two numbers

```
#include <iostream>
using namespace std;
int main()
{
    int n1, n2;
    cout << "Enter two numbers: ";
    cin >> n1 >> n2;
    while(n1 != n2)
    {
        if(n1 > n2)
            n1 -= n2;
        else
            n2 -= n1;
    }
    cout << "GDC = " << n1;
    return 0;
}
```

Output:

```
Enter two numbers: 78
52
GDC = 26
```

4. Write programs to find the sum of the following series:

(a) $X - \dfrac{X^2}{2!} + \dfrac{X^3}{3!} - \dfrac{X^4}{4!} + \dfrac{X^5}{5!} - \dfrac{X^6}{6!}$

```cpp
#include <iostream>
using namespace std;
int main()
{
int x,p,i,j;
double fact=1.0,ans=0;
cout<<"Enter the value of x:";
cin>>x;
cout<<"\n Enter till what power you want:";
cin>>p;
ans=x;
for(i=2,j=1;i<=p;i++,j++){
fact=fact*i;
if(i%2==0)
ans+=(pow(-1,j))*((pow(x,i))/(fact));
}
cout<<"\n The sum of the series is:"<<ans;
return 0;
}
```

OUTPUT:

Enter the value of x: 3
Enter till what power you want: 4
The sum of the series is: -4.875

(b) $x + \dfrac{x^2}{2} + \dfrac{x^3}{3} + ... + \dfrac{x^n}{n}$

```cpp
#include <iostream>
using namespace std;
int main()
{
clrscr();
int i,n;
float x,sum=0;
cout<<"x+x^2/2+x^3/3+…..+x^n/n";
cout<<"\n Enter value of x and n:";
cin>>x>>n;
for(i=1;i<=n;++i)
sum+=pow(x,i)/i;
cout<<"\n sum="<<sum;
getch();
}
```

1. **Write a program to find sum of the series**

$$S = 1 + x + x^2 + ... + x^n$$

```
#include <iostream>
using namespace std;
int main()
{
clrscr();
long i,n,x,sum=1;
cout<<"1+x+x^2+……+x^n";
cout<<"\n Enter the value of x and n:";
cin>>x>>n;
for(i=1;i<=n;++i)
sum+=pow(x,i);
cout<<"\n Sum="<<sum;
getch();
}
```

Output:

```
1+x+x^2+.......+x^n

Enter the value of x and n:5
2

Sum=31
```

Extra One Marks

1) Program statements that cause such jumps are called as **"Control flow"**.

2) The "**null** or **empty statement**" is a statement containing only a semicolon.

3) Group (b**races {}**) of statements is called as a **compound statement** or **a block.**

4) control statements are **three types**.

5) Sequential statements always end with a **semicolon (;)**

6) **Iteration statement** also known as **looping statement.**

7) iteration statements **three types**

8) The Conditional Operator is an alternative for **'if else Statement'.**

9) The **Switch** Statment is a **multi-way** branch statement.

10) In C++, any **nonzero is treated as true including negative numbers and zero is treated as false.**

11) The **if-else ladder** is a **multi-path decision making** statement.

12) The **conditional operator** that consists of two symbols (**?:**).

13) The **conditional operator** (or Ternary operator) is an alternative for **'if else statement'.**

14) The **switch statement** is a **multi-way** branch statement.

15) **Duplicate case values** are not allowed in **switch case**.

16) The **break statement** is used inside the switch to terminate a statement sequence.

17) **The if statement is more flexible than switch statement.**

18) The **switch statement** is more efficient than **if-else statement**.

19) A **switch statement** can only work for quality of comparisons.

20) **No two case labels** in the same **switch** can have identical values.

21) The **initialization expression** is executed only once in the beginning of the loop.

22) **Empty loop** means a loop has no statement in its body is called an empty loop.

23) **Jump statements** are used to interrupt the normal flow of program.

24) **Break** is used to **terminate** the execution of the loop.

Extra Questions (2 & 3 Marks)

1. **What are the kinds of statements used in C++.**
   There are two kinds of statements used in C++.
   - (i) Null statement
   - (ii) Compound statement

2. **Null statement**
   The "**null** or **empty statement**" is a statement containing only a semicolon.
   It takes the flowing form:
   ; // it is a null statement

3. **Compound (Block) statement**
   C++ allows a group of statements enclosed by pair of braces {}. This group of statements is called as a compound statement or a block.
   EX:
   ```
   {
   int x, y;
   x = 10;
   y = x + 10;
   }
   ```

4. **What is Control Statement?**
   - ➢ Control Statement are statements that alter the sequence of flow of instaructions.
   - ➢ If the Statements are executed sequentially, the flow is called as sequential flow. In some situations, if the statements alter the flow of execution like branching, iteration, jumping and function calls, this flow is called as control flow.

5. **What is Selection statement?**
   The selection statement means the statement (s) are executed depends upon a condition. If a condition is true, a true block (a set of statements) is executed otherwise a false block is executed. This statement is also called **decision statement** or **selection statement.**

6. **What are the kinds of control statement used in C++?**
   There are three kinds of control statement used in C++.
   - (1) Sequence Statement
   - (2) Selection Statement
   - (3) Iteration Statement

7. **Define: Iteration Statement (looping)**
   - ➢ Iteration Statement (looping) is use to execute a set of statements repeatedly until a condition is satisfied.
   - ➢ If a condition evaluates to true, the set of statements (true block) is executed again and again.
   - ➢ This is also known as **looping statement** or iteration statement.

8. **What are the kinds Iteration Statements supported in C++?**
   There are three kinds Iteration Statements supported.
   - (1) *for*
   - (2) *While*
   - (3) *do-While.*

9. **What are the kinds of Jump statement used in C++?**
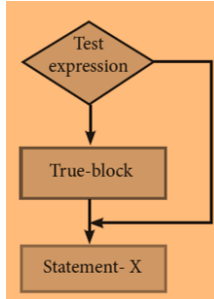   In C++ three Jump Statment are used
   - (1) *goto*
   - (2) *break*
   - (3) *continue*

10. What is if statement?

The if statement evaluates a condition, if the condition is true then a true-block (a statement or set of statements) is executed, otherwise the true-block is skipped.

The general syntax of the if statement is:

    if (expression)
    true-block;
    statement-x;

11. Draw the flow chart of if-statement.

12. Write the syntax of the if-else statement?

    if (expression)
    {
    True-block;
    }
    else
    {
    False-block;
    }
    Statement-x

13. What are the nested can have one of the three forms?
1. If nested inside if part
2. If nested inside else part
3. If nested inside both if part and else part

14. Define Switch statement?

The switch statement is a multi-way branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression. The switch statement replaces multiple if-else sequences.

15. What are the parts used to loop?

Every loop has four elements that are used for different purposes. These elements are
- Initialization expression
- Test expression
- Update expression
- The body of the loop

16. Define: for loop

The for loop is the easiest looping statement which allows code to be executed repeatedly.

The general syntax is:

    for (initialization(s); test-expression; update expression(s))
    {
    Statement 1;
    Statement 2;
    ………….
    }

17. What is Empty loop?

Empty loop means a loop has no statement in its body is called an empty loop.
Following for loop is an empty loop:

for(i+0   ;   i<=5; +=i)(); ←———— The body of for loop
                                                   contains a null statement

18. Define: While loop

A while loop is a control flow statement that allows the loop statements to be executed as long as the condition is true.

*The while loop syntax is***:**

while (Test expression)
{
Body of the loop;
}
Statement-x;

19. Explain: do-while loop

The do-while loop is an exit-controlled loop. In do-while loop, the condition is evaluated at the bottom of the loop after executing the body of the loop.

The do-while loop syntax is:

do
{
Body of the loop;
}
while(condition);

20. What is Jump statement?

Jump statements are used to interrupt the normal flow of program.
Types of Jump Statements are
✓ goto statement
✓ break statement
✓ continue statement

21. Define goto statement.

The goto statement is a control statement which is used to transfer the control from one place to another place without any condition in a program.

22. What is break statement?

A break statement is a jump statement which terminates the execution of loop and the control is transferred to resume normal execution after the body of the loop.

23. Difference between Break and Continue

Break	Continue
Break is used to **terminate** the execution of the loop.	Continue is not used to terminate the execution of loop.
It **breaks** the iteration.	It **skips** the iteration.
When this statement is executed, control will come out from the loop and executes the statement immediate after loop.	When this statement is executed, it will not come out of the loop but moves/jumps to the next iteration of loop.
Break is used with loops as well as switch case.	Continue is only used in loops, it is not used in switch case.

Extra Questions (5 Marks)

1. **Define Selection statements. Explain its Types.**

The selection statement means the statement (s) are executed depends upon a condition. If a condition is true, a true block (a set of statements) is executed otherwise a false block is executed. This statement is also called **decision statement** or **selection statement.**

Types of selection statements:
   a) If
   b) If else
   c) Nested if
   d) Else if ladder
   e) Conditional operator
   f) Switch case

a) if statement

The if statement evaluates a condition, if the condition is true then a true-block (a statement or set of statements) is executed, otherwise the true-block is skipped.

Syntax:

```
if (expression)
true-block;
statement-x;
```
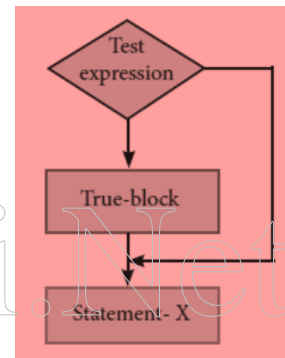
Flow chart:

Example:

```
#include<iostream>
using namespace std;
int main()
{
int a=10, b=5;
if(a>b)
{
cout<<a;
}
getch();
}
```
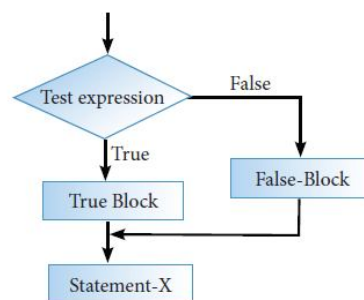
Output:

   10

b) If else

The if statement evaluates a condition, if the condition is true execute first statement, otherwise (false) execute second statement.

Syntax:

```
if ( expression)
{
True-block;
}
else
{
False-block;
}
Statement-x
```

Flow chart:

Example:

```
#include<iostream>
using namespace std;
int main()
{
int a=10, b=5;
if(a>b)
cout<<a;
else
cout<<b;
getch();
}
```

Output:

   5

c) Nested if

An if statement contains another if statement is called nested if. The nested can have one of the following three forms.

1. If nested inside if part
2. If nested inside else part
3. If nested inside both if part and else part

d) if -else-if ladder

The if-else ladder is a multi-path decision making statement. In this type of statement 'if' is followed by one or more else if statements and finally end with an else statement.

Syntax:　　　　　　　　　　　　　　　　　**Example:**

```
if (expression 1)
{
Statemet-1
}
else
if( expression 2)
{
Statemet-2
}
else
if ( expression 3)
{
Statemet-3
}
else
{
Statement-4
}
```

```
#include <iostream>
using namespace std;
int main ()
{
int marks;
cout<<" Enter the Marks :";
cin>>marks;
if( marks >= 60 )
cout<< "Your grade is 1st class !!" <<endl;
else if( marks >= 50 && marks < 60)
cout<< "your grade is 2nd class !!" <<endl;
else if( marks >= 40 && marks < 50)
cout<< "your grade is 3rd class !!" <<endl;
else
cout<< "You are fail !!" <<endl;
return 0;
}
```
Output
```
Enter the Marks :60
Your grade is 1st class !!
```

e) The ?: Alternative to if- else

The conditional operator (or Ternary operator) is an alternative for 'if else statement'. The conditional operator that consists of two symbols (?:). It takes three arguments.

Syntax:

expression 1? expression 2 : expression 3;

Example:

```
#include <iostream>
using namespace std;
int main()
{
int a=10, b=5;
if(a>b ? cout<<a : cout<<b)
getch();
}
```
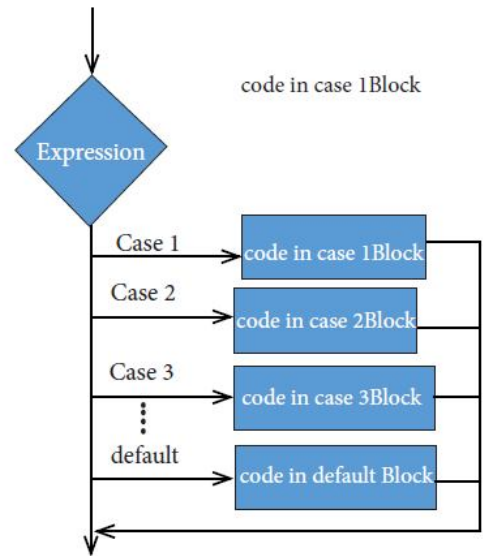**Output:**

10

2. Explain switch case statement with example program?

The switch statement is a multi-way branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.

Syntax:

```
switch(expression)
{
case constant 1:
statement(s);
break;
case constant 2:
statement(s);
break;
  .
  .
  .
default:
statement(s);
     }
```


Rules:

1. The expression provided in the switch should result in a constant value otherwise it would not be valid.
2. Duplicate case values are not allowed.
3. The default statement is optional.
4. The break statement is used inside the switch to terminate a statement sequence. When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
5. The break statement is optional. If omitted, execution will continue on into the next case. The flow of control will fall through to subsequent cases until a break is reached.
6. Nesting of switch statements is also allowed.

Example:

```
#include <iostream>
using namespace std;
int main()
{
int num;
cout << "\n Enter week day number: ";
cin >> num;
switch (num)
{
case 1 : cout << "\n Sunday"; break;
case 2 : cout << "\n Monday"; break;
case 3 : cout << "\n Tuesday"; break;
case 4 : cout << "\n Wednessday"; break;
case 5 : cout << "\n Thursday"; break;
case 6 : cout << "\n Friday"; break;
case 7 : cout << "\n Saturday"; break;
default: cout << "\n Wrong input....";
}
getch();
}
```

Output:

```
        Enter week day number: 6
                Friday
```

3. What is Jump statement? Explain its types with example.

Jump statements are used to interrupt the normal flow of program.

Types of Jump Statements are

1. *goto statement*
2. *break statement*
3. *continue statement*

1. goto statement

The goto statement is a control statement which is used to transfer the control from one place to another place without any condition in a program.

syntax :

| Syntax1	Syntax2
goto label;	label:
--------------	-------------
--------------	-------------
--------------	-------------
label:	goto label;

```
#include<iostream.h>
void main()
{
cout << "\n Statement 1.";
cout << "\n Statement 2.";
cout << "\n Statement 3.";
goto last;
cout << "\n Statement 4.";
last:
cout << "\n End of Program.";
}
```
Output :
```
Statement 1.
Statement 2.
Statement 3.
End of Program.
```

2. break statement

A break statement is a jump statement which terminates the execution of loop and the control is transferred to resume normal execution after the body of the loop.
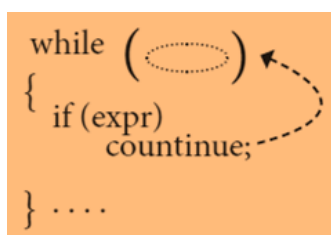
syntax :
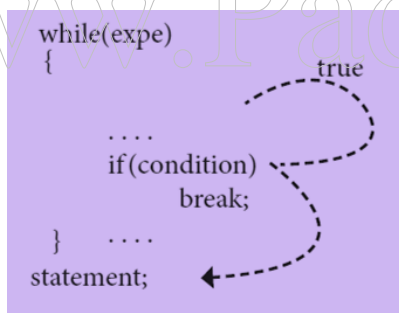

```
#include<iostream.h>
void main()
{
int a=1;
while(a<=10)
{
if(a==3)
break;
cout << "\nStatement " << a;
a++;
}
cout << "\nEnd of Program.";
}
```
Output :
```
Statement 1
Statement 2
End of Program.
```

3. continue statement

The continue statement works quite similar to the break statement. Instead of terminating the loop (break statement), continue statement forces the loop to continue or execute the next iteration.

syntax :


```
#include<iostream.h>
void main()
{
int a=0;
while(a<5)
{
a++;
if(a==3)
continue;
cout << "\nStatement " << a;
}
cout << "\nEnd of Program.";
}
```
Output :
```
Statement 1
Statement 2
Statement 4
Statement 5
End of Program.
```

4. Write Key Differences Between if-else and switch.

1. Expression inside if statement decide whether to execute the statements inside if block or under else block. On the other hand, expression inside switch statement decide which case to execute.

2. An if-else statement uses multiple statements for multiple choices. On other hand, switch statement uses single expression for multiple choices.

3. If-esle statement checks for equality as well as for logical expression. On the other hand, switch checks only for equality.

4. The if statement evaluates integer, character, pointer or floating-point type or Boolean type. On the other hand, switch statement evaluates only character or a integer data type.

5. Sequence of execution is like either statement under if block will execute or statements under else block statement will execute. On the other hand the expression in switch statement decide which case to execute and if do not apply a break statement after each case it will execute till the end of switch statement.

6. If expression inside if turn out to be false, statement inside else block will be executed. If expression inside switch statement turn out to be false then default statements are executed.

7. It is difficult to edit if-else statements as it is tedious to trace where the correction is required. On the other hand, it is easy to edit switch statements as they are easy to trace.

*****

M.VIJAYA KUMAR, MCA.,M.Phil.,B.Ed.,PGDCA.,   **V.SUJITHA, MCA.,M.Phil.,**

PGT-COMPUTER TEACHER,   PGT-COMPUTER TEACHER,

SASTRA MATRICULATION HIGHER SECONDARY SCHOOL,
KILPENNATHUR, TIRUVANNAMALAI 604601

Mobile No : 9655826843   *Email : vijay28soft@gmail.com*

ALL THE BEST