FAST

National University of Computer and Emerging Sciences Peshawar

OOP Lab # 3.1

DEPARTMENT OF COMPUTER SCIENCE

C++ Programming (Conditional Statements)

Computer Instructor: Muhammad Abdullah Orakzai

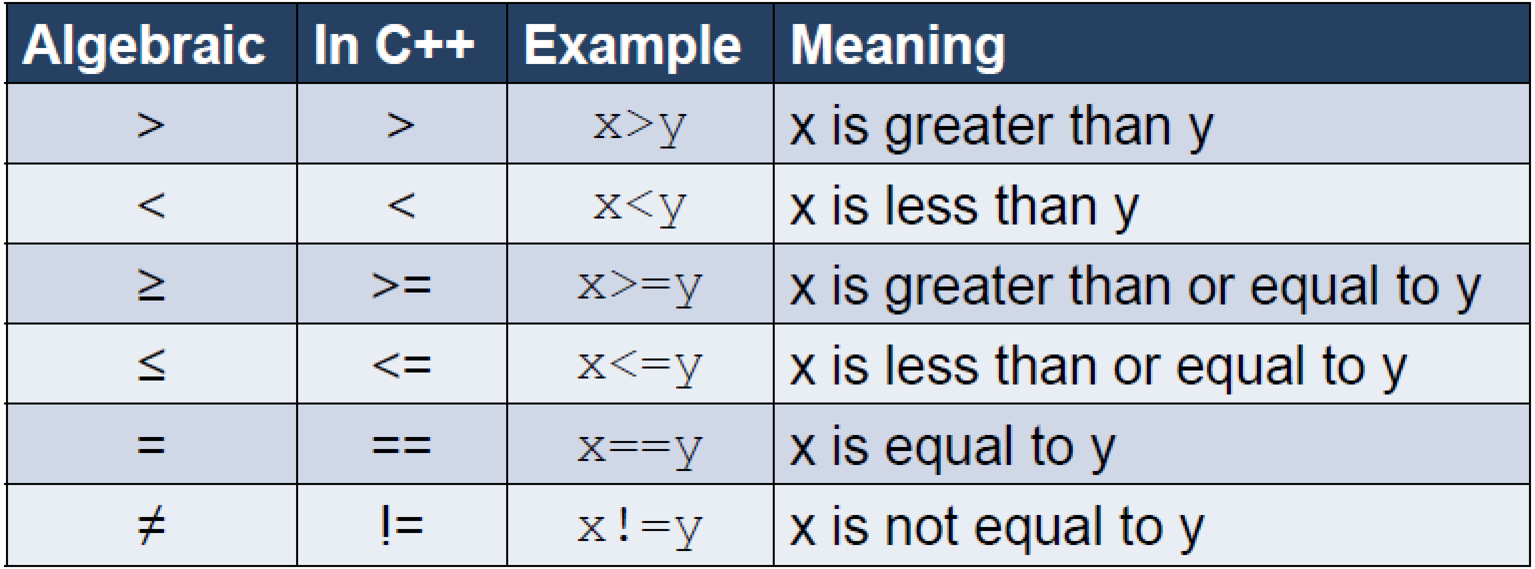
Programming

Prepared By: Muhammad Abdullah Orakzai (Lab Instructor CS)

Contents

1. if statement
2. if-else statement
3. if-else if- else statement
4. Nested if-else statement
5. Conditional operator
6. Switch Statement
7. Boolean variable

Relational Operators/comparison operators



Algebraic	In C++	Example	Meaning
>	>	`x>y`	x is greater than y
<	<	`x<y`	x is less than y
≥	>=	`x>=y`	x is greater than or equal to y
≤	<=	`x<=y`	x is less than or equal to y
=	==	`x==y`	x is equal to y
≠	!=	`x!=y`	x is not equal to y

Blocks of Code

- Whenever we write an if statement or a loop, if there is more than one statement of code which has to be executed, this has to be enclosed in braces, i.e. ‘{ …. }’

Conditional Statements

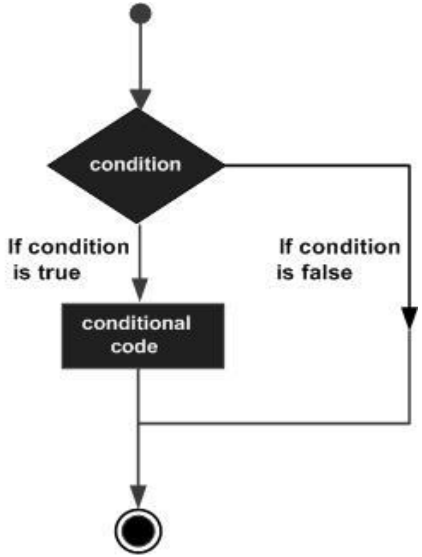
- Also called decision making statements decision control statements.
- Decision making structures have one or more conditions to be evaluated or tested by the program, along with a statement or statements that are to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.
- Used to change the flow.
- In these statements (conditions) order or sequence of the statements are changed.
- Sometimes we need to execute a block of statements only when a particular condition is met or not met. This is called **decision making**, as we are executing a certain code after making a decision in the program logic.

Conditional Statements…

1. If Statement
2. If-else Statement
3. If-else if- else statement
4. Switch statement
5. Nested Statement

- Use if to specify a block of code to be executed, if a specified condition is true
- Use else to specify a block of code to be executed, if the same condition is false
- Use else if to specify a new condition to test, if the first condition is false
- Use switch to specify many alternative blocks of code to be executed

Conditional Statements…



1) if Statement

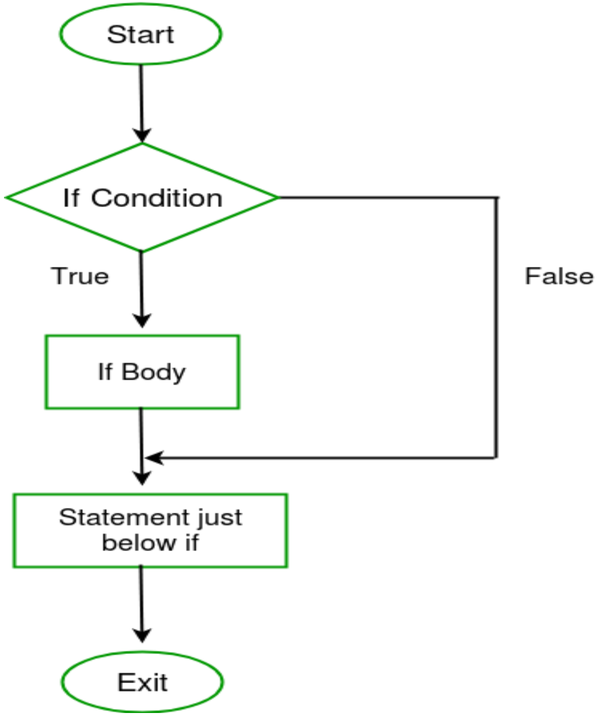
if statement will execute or skip or ignore a block of code depending one condition

Syntax:

```
if(conditional expression)
{

 // Statements will execute if the Boolean expression is true


}
```

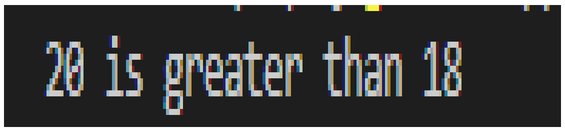
The statements inside **if** parenthesis (usually referred as if body) gets executed only when the given condition is true. If the condition is false than the statements inside if body are completely ignored.

1) if Statement…



1) if Statement...

```
#include<iostream>
using namespace std;
int main()
{
   if (20>18)
   {
       cout<<"20 is greater than 18";
   }


}
```



```
20 is greater than 18
```

1) if Statement…

We can also test variables:

```
#include <iostream>

using namespace std;

int main() {

  int x = 20;

  int y = 18;

  if (x > y) {

    cout << "x is greater than y";

  }

  return 0;

}
```

2) if else Statement

❖Used for making two way decision.

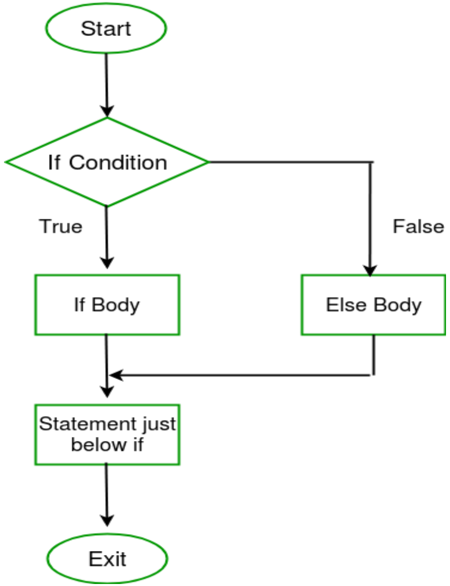
❖It will execute if block if condition is true and will execute another block (else block) if condition is false.

❖It will take one action if the condition is true and take another action if condition is false.

❖Sometimes you have a condition and you want to execute a block of code if condition is true and execute another piece of code if the same condition is false. This can be achieved in C++ using if-else statement.
This is how an if-else statement looks:

2) if else Statement…

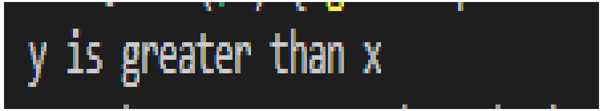
```
if(condition)
{
Statement(s);
}else
{
Statement(s);
}
```



The statements inside “if” would execute if the condition is true, and the statements inside “else” would execute if the condition is false.

2) if else Statement…Example

```
#include<iostream>
using namespace std;
int main()
{   int x=15;
    int y=18;

   if (x>y)
   {
       cout<<"x is greater than y";
   }
   else
   {
       cout<<"y is greater than x";
   }
return 0;

}
```



```
y is greater than x
```

if else Statement Tasks

1. Find positive and negative numbers using if else statement.
2. Find even and odd numbers using if else statement.
3. Find leap year using if else statement.
4. Write a C++ program which will get two numbers from user and find large number between them using if else statement.

Leap year Hints: common year has 365 days (feb 28 days)

Leap year has 366 days (feb 29 days)

year%4==0   leap year

3) if-else-if else Statement

- This statement is used to check multiple conditions.
- Used to execute one condition from multiple statements.
- if-else-if statement is used when we need to check multiple conditions. In this control structure we have only one “if” and one “else”, however we can have multiple “else if” blocks. This is how it looks:

3) if-else-if else Statement...

```
if(condition_1)
{
/*if condition_1 is true execute this*/
statement(s);
}else if(condition_2)
{
/*execute this if condition_1 is not met and condition_2 is met*/
statement(s);
}
```

3) if-else-if else Statement...

```
else if(condition_3)
{   /* execute this if condition_1 & condition_2 are    * not met
and condition_3 is met    */
statement(s);
}
.
.
.
```

3) if-else-if else Statement…

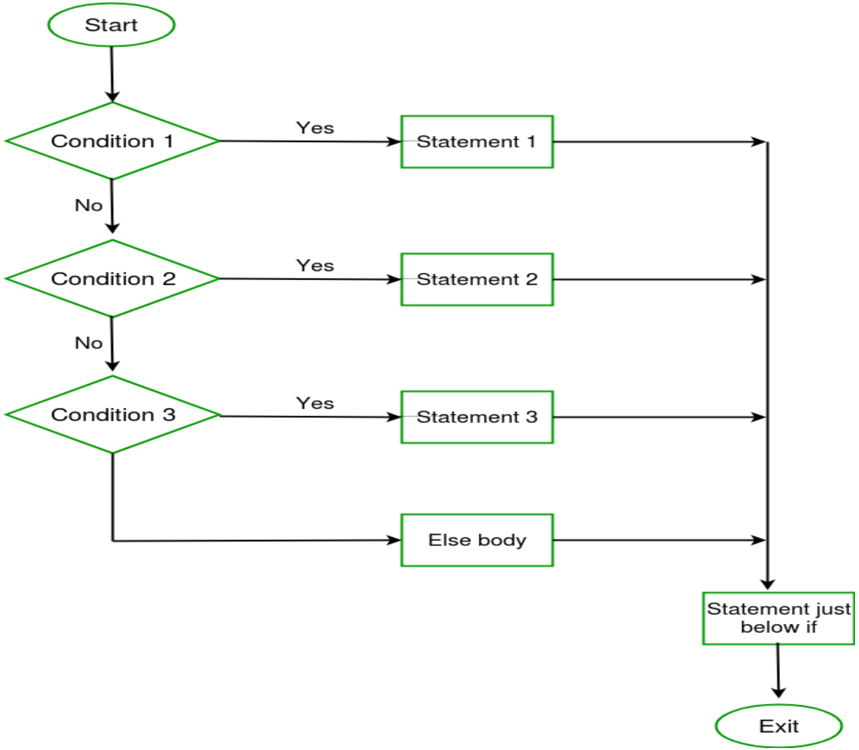
```
else {

/* if none of the condition is true then these statements gets
executed    */

statement(s);

}
```

3) if-else-if else Statement…

Note:

- The most important point to note here is that in if-else-if, as soon as the condition is met, the corresponding set of statements get executed, rest gets ignored.
- If none of the condition is met then the statements inside “else” gets executed.

3) if-else-if else Statement…

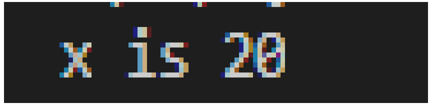


3) if-else-if else Statement…

```
#include<iostream>
using namespace std;

int main()
{
    int x=20;

   if (x==10)
   {
       cout<<"x is 10";
   }
   else if(x==15)
   {
       cout<<"x is 15";
   }
```

3) if-else-if else Statement...

```
else if(x==20)
   {
       cout<<"x is 20";
   }
else
   {
            cout<<"sorry! x is not present";
   }

return 0;

}
```



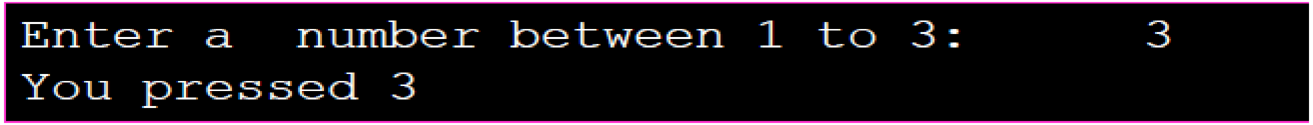
```
x is 20
```

3) if-else-if else Statement...

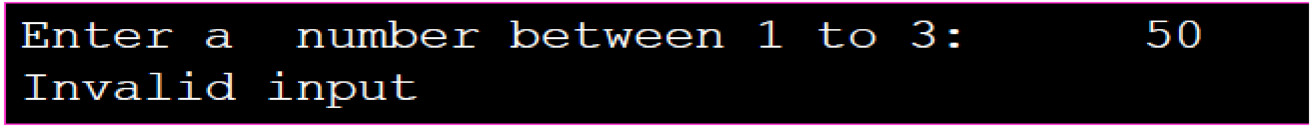
```
#include<iostream>
using namespace std;
int main()
{
int number;
cout<<"Enter a number between 1 to 3:";
cin>> number;
if(number == 1)
{
cout<<"You pressed 1" <<endl;
}
else if(number ==2)
{
cout<<"You pressed 2" <<endl;
}
```

3) if-else-if else Statement…

```
else if(number == 3)
{
cout<<"You pressed 3" <<endl;
}
else
{
cout<<"Invalid input";
}
}
```



```
Enter a  number between 1 to 3:      3
You pressed 3
```



```
Enter a  number between 1 to 3:      50
Invalid input
```

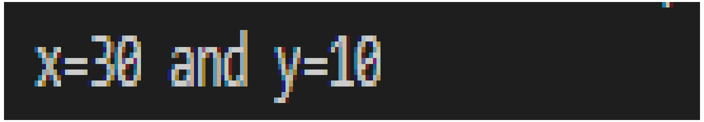
if-else-if else Statement Tasks

1. Find positive, negative and neutral numbers using multiple if statement.
2. Take value of temperature from user and find status of weather accordingly.
3. Take value of percentage from user and find grades based on percentage value.
4. Make a calculator using if-else-if else statement which perform the addition, subtraction, multiplication, division and remainder operations. Take values and operator from user on runtime.

4) Nested if statement

You can use one **if** or **else if** statement inside another **if** or **else if** statement(s).

```
#include<iostream>
using namespace std;
int main()
{
    int x=30;
    int y=10;
   if (x==30)
   {
       if(y==10)
       {
           cout<<"x=30 and y=10";
       }
   }
return 0;
}
```



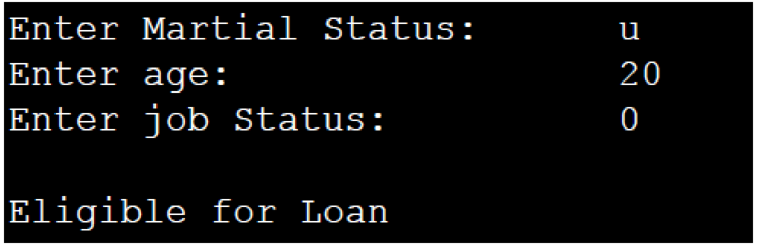
```
x=30 and y=10
```

4) Nested if statement…

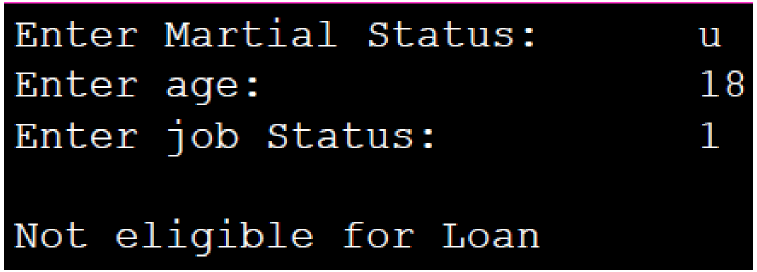
```
#include<iostream>
using namespace std;

int main()
{
bool job; char martialStatus; int age;

cout<<"Enter Martial Status:";
cin>>martialStatus;

cout<<"Enter age:";
cin>>age;

cout<<"Enter job Status:";
cin>>job;
```

4) Nested if statement…

```
if(martialStatus =='u')
{
    if(age <=25)
        if(job ==false)
            cout<<"Eligible for Loan";
}

else
    cout<<"Not eligible for Loan";
return 0;

}
```



```
Enter Martial Status:        u
Enter age:                   20
Enter job Status:            0

Eligible for Loan
```



```
Enter Martial Status:        u
Enter age:                   18
Enter job Status:            1

Not eligible for Loan
```

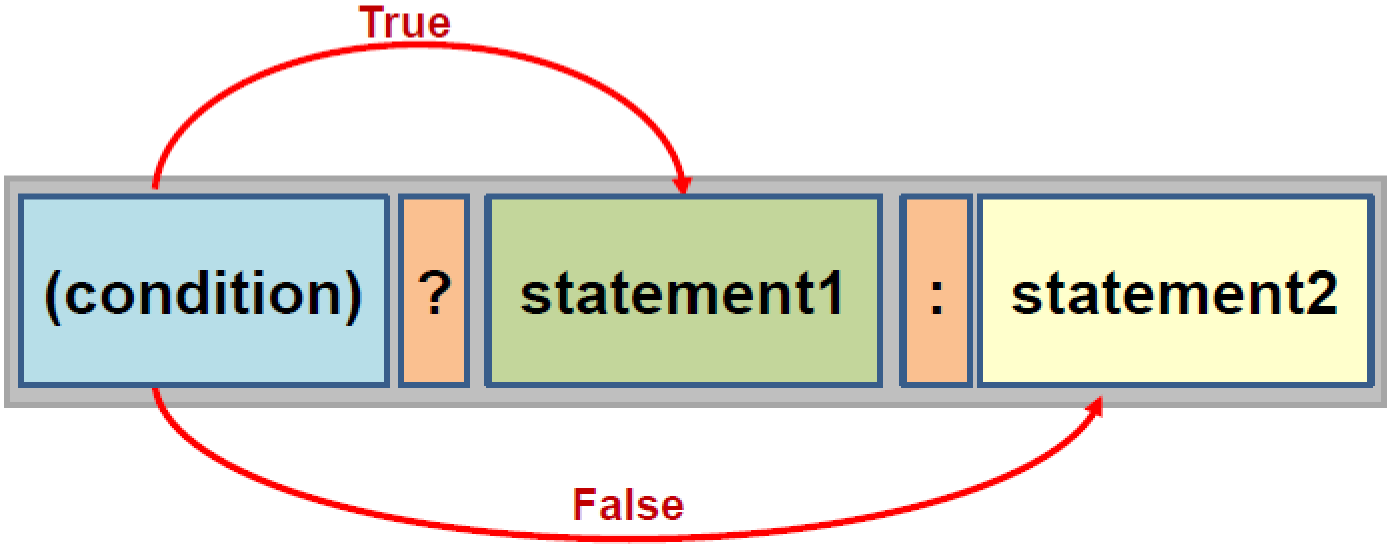
5) Conditional Operator (? :)

- It is ternary operator and work on three operands.
- It works like if else statement.
- **Syntax:**

```
(condition)   ? statement 1 : statement 2;
```

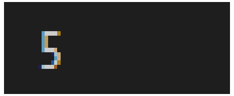
Statement must be single and it is the limitation of conditional operator.

5) Conditional Operator (? :)...



5) Conditional Operator (? :) Example

```
#include<iostream>
using namespace std;

int main()
{
    int n1=7;
    int n2=5;
    int result = (n1>n2) ? n1 : n2;
    cout<<result;
return 0;

}
```



```
5
```

Conditional Operator (? :) Tasks

1. Write a C++ program which will get two numbers from user and find large number between them using conditional operator.
2. Find positive and negative numbers using conditional operator.
3. Find even and odd numbers using conditional operator.

Logical Operators

Used for compound condition or expression

1) AND (&&)
2) OR (||) pip sign

1) AND (&&)

```
#include<iostream>
using namespace std;
int main()
{
    int x=30;
    int y=10;

   if (x==30 && y==10)
   {
           cout<<"x=30 and y=10";
   }
   else
   {
    cout<<"Either x is not equal to 30 or y is not e
qual to 10 or both are not equal";
   }
return 0;
}
```

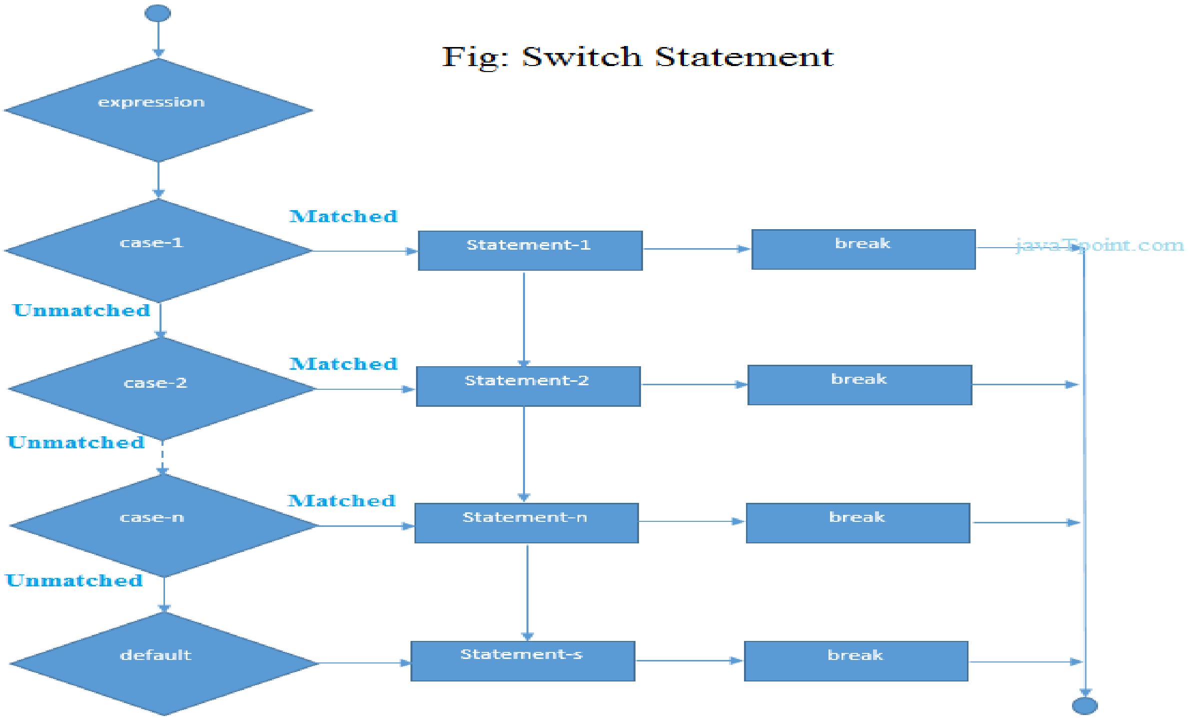
1) OR (||)

```
#include<iostream>
using namespace std;
int main()
{
    int x=30;
    int y=10;

   if (x==30 || y==10)
   {
           cout<<"x=30 or y=10";
   }
   else
   {
    cout<<"Both are not equal means x is not equal to 30 and y
is not equal to 10";
   }
return 0;

}
```

6) Switch statement

- Used when multiple choices are given and one is to be selected. It is like if else-if- if statement.
- Used to select one several actions based on the value of variable or expression.
- **Switch case statement** is used when we have number of options (or choices) and we may need to perform a different task for each choice.

6) Switch statement…



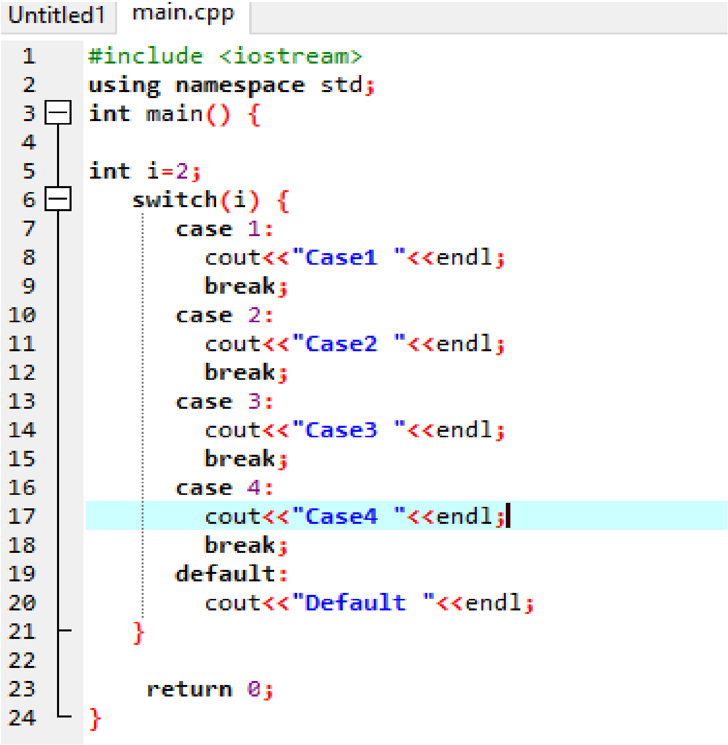
6) Switch statement Syntax

```
switch(variable/expression)
{
case value 1:
statement(s);   // code to be executed
break;
case value 2:
statement(s);   // code to be executed
break;
```

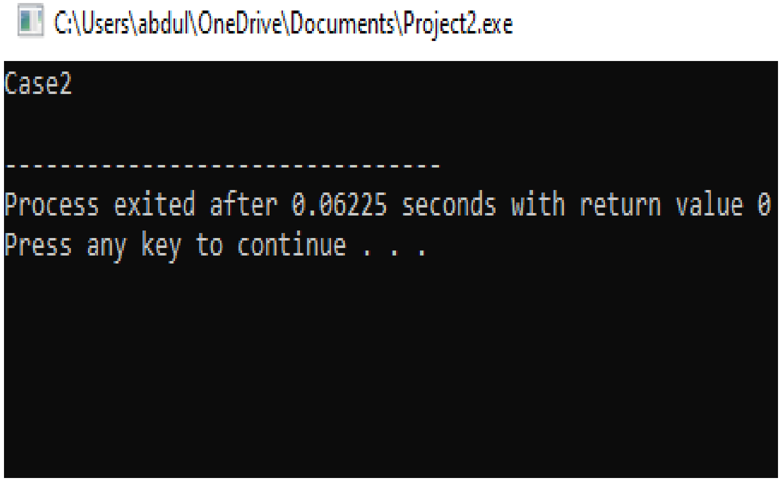
6) Switch statement Syntax…

```
.
.
.
case value n:
Statement(s);     // code to be executed
break;
default:
statement(s);     // code to be executed if all cases are not matched
}
```

6) Switch statement Example



```
#include <iostream>
using namespace std;
int main() {

int i=2;
   switch(i) {
      case 1:
        cout<<"Case1 "<<endl;
        break;
      case 2:
        cout<<"Case2 "<<endl;
        break;
      case 3:
        cout<<"Case3 "<<endl;
        break;
      case 4:
        cout<<"Case4 "<<endl;
        break;
      default:
        cout<<"Default "<<endl;
   }

    return 0;
}
```



```
Case2


--------------------------------
Process exited after 0.06225 seconds with return value 0
Press any key to continue . . .
```

Switch statement Tasks

1. Make a C++ calculator using switch statement which perform the following addition, subtraction, multiplication, division and remainder value. Take value and operator from user on runtime.
2. Write a C++ program using switch statement which get month name from user and display month number accordingly.

Switch statement VS if else if

❖If a program contains conditions or compound conditions then we use if else if else

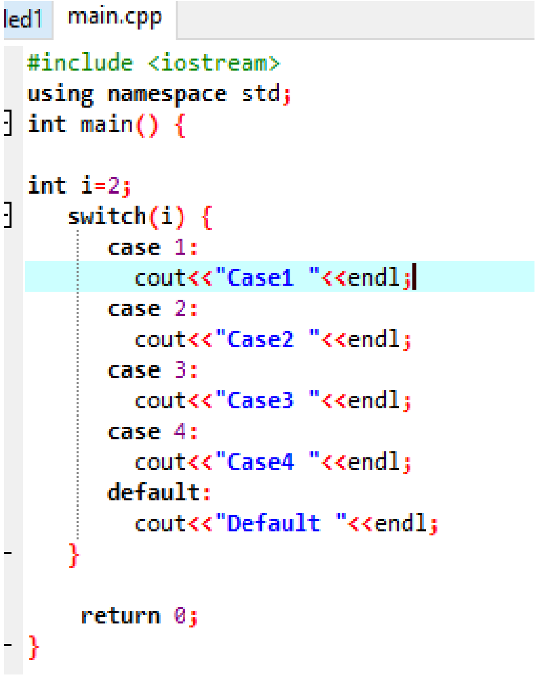
❖If program contains single variable or expression then we use switch statement.

Compound conditions

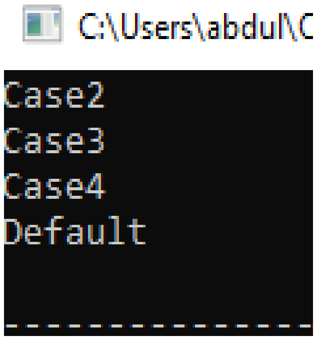
1) if(a>b && a>c)
2) if)(a>b || a>c)

C++ Switch statement is fall through

It means it executes all statements after match if break statement is not used with switch cases.



```cpp
#include <iostream>
using namespace std;
int main() {

int i=2;
    switch(i) {
        case 1:
          cout<<"Case1 "<<endl;
        case 2:
          cout<<"Case2 "<<endl;
        case 3:
          cout<<"Case3 "<<endl;
        case 4:
          cout<<"Case4 "<<endl;
        default:
          cout<<"Default "<<endl;
    }

     return 0;
}
```



```
Case2
Case3
Case4
Default

---------------
```

Java Switch statement is fall through...

```
#include <iostream>
using namespace std;
int main() {
cout<<"Enter alphabet: ";
char alphabet;
cin>>alphabet;
switch(alphabet)
{
    case 'a':
    case 'e':
    case 'i':
    case 'o':
```

Java Switch statement is fall through...

```
    case 'u':
    case 'A':
    case 'E':
    case 'I':
    case 'O':
    case 'U':
    cout<<"You entered vowel";
    break;
    default:
    cout<<"You entered consonant";
}   // switch body Closed
return 0;
}
```

Java Switch statement is fall through…

```
            case 'O' :
            case 'U' :
            cout<<"You entered vowel";
            break;
            default:
            cout<<"You entered consonant";
}   // switch body Closed
return 0;
}
```

```
Enter alphabet: a
You entered vowel

 Enter alphabet: I
 You entered vowel

 Enter alphabet: s
 You entered consonant
```

Break Statement

The break statement is used to exit from the body of the switch structure or loop structure.

The break statement terminates the execution of the loop when it is used inside the body of the loop.

Syntax: break;

boolean variable

Syntax:

bool variable_name;

e.g: bool even;

boolean variable example

```
#include <iostream>
using namespace std;
int main() {
cout<<"Enter any number: ";
int number;
cin>>number;
bool even;   // bool variable declaration
even =(number % 2==0);
if(even)
{
cout<<"Even number";
}
else
{
cout<<"Odd Number";
}
return 0;
}
```

```
Enter any number: 6
Even number

Enter any number: 7
Odd Number
```

Summary

Sr.No	Statement & Description
1	if statement An ‘if’ statement consists of a Boolean expression followed by one or more statements.
2	if...else statement An ‘if’ statement can be followed by an optional ‘else’ statement, which executes when the Boolean expression is false.

Summary…

3	if...else if...else statement An ‘if’ statement can be followed by one or multiple else if statement and none of above is true then ‘else’ statement, which executes when the Boolean expression is false.
4	switch statement A ‘switch’ statement allows a variable to be tested for equality against a list of values.
5	nested statements You can use one statement inside another statement(s).

References

- https://beginnersbook.com/2017/08/cpp-data-types/
- https://www.geeksforgeeks.org/c-data-types/
- http://www.cplusplus.com/doc/tutorial/basic_io/
- https://www.geeksforgeeks.org/basic-input-output-c/
- https://www.w3schools.com/cpp/default.asp
- https://www.javatpoint.com/cpp-tutorial

THANK YOU