Conditional Statement

Relational operators   (< > <= and >=)

The relational operators are used to compare relative values.

E1 < E2   Gives 1 (true) if the value of E1 is less than value of E2; otherwise, the result is 0 (false)

E1 <= E2   Gives 1 (true) if the value of E1 is less than or equal to the value of E2; otherwise, the result is 0 (false).

E1 > E2   Gives 1 (true) if the value of E1 is greater than the value of E2;   otherwise, the result is 0 (false)

E1 >= E2   Gives 1 (true) if the value of E1is greater than or equal to the   value of E2; otherwise, the result is 0 (false).

Equality operators   (== and !=)

The operators == and != are used to test for equality or inequality between arithmetic or pointer values, following rules similar to those for the relational operators.

The equality operators have lower precedence than the relational operators, however, and you can also compare certain pointer types not allowed with relational operations.

If E1 == E2 gives 1 (true), then either E1 and E2 point to the same function, or they are both null.

The expression E1 != E2 follows the same rules, except that the result is 1 (true) if the operands are unequal, and 0 (false) if the operands are equal.

The if statement (keyword)

The first type of branching statement we will look at is the **if statement**. The if statement can cause other statements to execute only under certain conditions. An **if statement** has the form:

if (<expression>) <statement1>;

If <expression> is non-zero when evaluated, <statement1> is executed.

Example:
 if (count < 50) count++;

Here is a simple C++ program demonstrating an if statement:

```
#include <iostream.h>

int main()
{
    int b;
    cout <<"Enter a value:";
    cin >> b;
    if (b < 0)
        cout <<"The value is negative";
    return 0;
}
```

This program accepts a number from the user. It then tests the number using an if statement to see if it is less than 0. If it is, the program prints a message "The value is negative". Otherwise, the program is silent. The **(b < 0)** portion of the program is the <u>Boolean expression</u>. C++ evaluates this expression to decide whether or not to print the message. If the Boolean expression evaluates to **True**, then C++ executes the single line immediately following the if statement (or a block of lines within braces immediately following the if statement). If the Boolean expression is **False**, then C++ skips the line or block of lines immediately following the if statement.
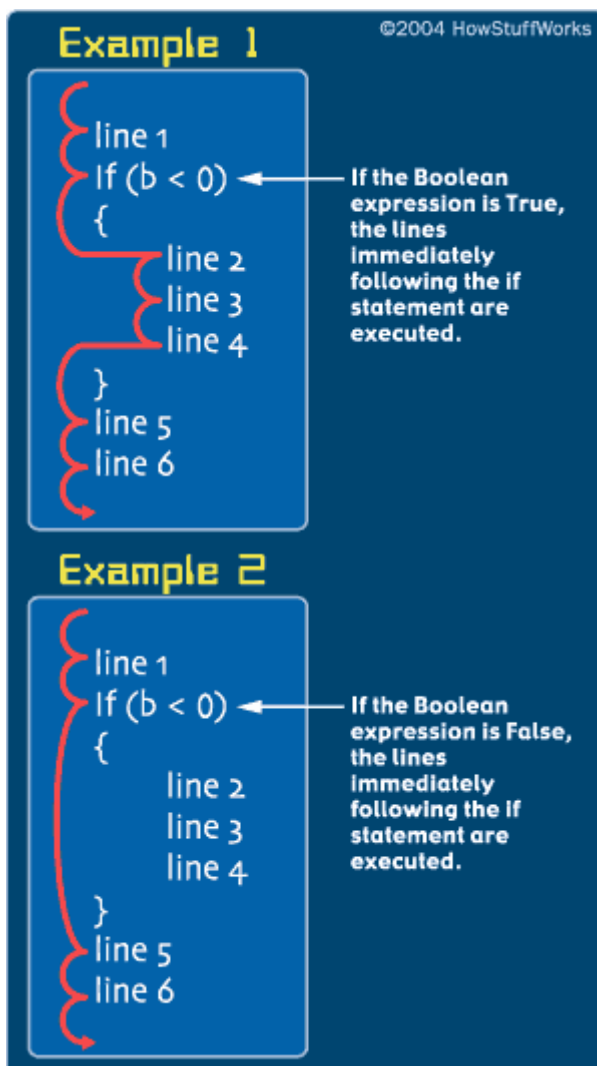
If (condition)

{

 Statement1;

Statement2;

.

.

Statementn;

}

Example: to update the previous program, if the value of b is negative then add 50 to b then print the new value of b. The program will be:

```cpp
#include <iostream.h>

int main()
{
    int b;
    cout << "Enter a value:";
    cin >> b;
    if (b < 0)
        {
         b+=50;
        cout <<"b="<<b;
        }
    return 0;
}
```

Q) Write a program to check if the no. is positive, then print positive number

```cpp
#include <iostream.h>
#include <conio.h>
int main()
{
int x;
clrscr();
cout << "Number:=";
cin >> x;
 if (x > 0)
  cout << " --- Positive Number ---";
cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to check if the student pass in English language

```cpp
#include <iostream.h>
#include <conio.h>
int main()
{
int x;
clrscr();
cout << "Degree:=";
cin >> x;
 if (x >=50)
  cout << " --- Pass ---";
cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to check the no., if it is even then print even number

```cpp
#include <iostream.h>
#include <conio.h>
int main()
{
int x;
clrscr();
cout << "Number:=";
cin >> x;
 if ((x % 2)==0)        // if (!(x % 2))
  cout << " --- Even Number ---";
cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to check if the no. accept division by 5 without remainder, then print that

```cpp
#include <iostream.h>
#include <conio.h>
int main()
{
int x;
clrscr();
cout << "Number:=";
cin >> x;
 if ((x % 5)==0)          // if (!(x % 5))
   cout << " --- Number accept division by 5 ---";
cout << "\n\nHit any key to continue";
getch()
return 0;
}
```

Q) Write a program to solve the following equation

y=x+1-y   when x>=20

```cpp
#include <iostream.h>
#include <conio.h>
int main()
{
int x,y;
clrscr();
cout << "x:=";
cin >> x;
cout << "y:=";
cin >> y;

 if (x >=20)
   {
   y=x+1-y;
   cout << " y:="<<y;
   }
cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

The If –else statement

□ if (<expression>) <statement1>;
  else <statement2>;

If <expression> is non-zero when evaluated, <statement1> is executed. Otherwise <statement2> is executed if the expression is 0.

An optional else can follow an if statement, but no statements can come between an if statement and an else.

Example:
 if (x < y)
  z = x;
 else
  z = y;

The #if and #else preprocessor statements (directives) look similar to the if and else statements, but have very different effects.

They control which source file lines are compiled and which are ignored.

- In an **if statement**, *condition* is a value or an expression that is used to determine which code block is executed, and the curly braces act as "begin" and "end" markers.

if (*condition*)
{
  // code to execute if *condition* is true
}
else
{
  // code to execute if *condition* is false
}

Here is a full C++ program as an example:

```
#include <iostream.h>  //include this file for cout
int main() {
  // define two integers
  int x = 3;
  int y = 4;

  //print out a message telling which is bigger
  if (x > y)
    cout << "x is bigger than y" << endl;

  else
    cout << "x is not bigger than y" << endl;

  return 0;
}
```

In this case *condition* is equal to "(x > y)" which is equal to "(3 > 4)" which is a *false* statement. So the code within the **else** clause will be executed. The output of this program will be:

```
x is not bigger than y
```
If instead the value for x was 6 and the value for y was 2, then *condition* would be "(6 > 2)" which is a *true* statement and the output of the program would be:
```
x is bigger than y
```

Q) Write a program to check if the no. is positive or not positive

```cpp
#include <iostream.h>
#include <conio.h>
int main()
}
int x;
clrscr();
cout << "Number:=";
cin >> x;
 if (x > 0)
  cout << " --- Positive Number ---";
 else
  cout << " --- The number is not positive ---";

cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to check if the student pass in English language or not

```cpp
#include <iostream.h>
#include <conio.h>
int main()
}
int x;
clrscr();
cout << "Degree:=";
cin >> x;
 if (x >=50)
  cout << " --- Pass ---";
else
  cout << " --- Fail ---";

cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to check if the NO. is even or odd

```
#include <iostream.h>
#include <conio.h>
int main()
}
int x;
clrscr();
cout << "Number:=";
cin >> x;
 if ((x % 2)==0)        // if (!(x % 2))
  cout << " --- Even Number ---";
else
  cout << " --- Odd Number ---";

cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to check if the no. accept division by 5 without remainder, then print that

```
#include <iostream.h>
#include <conio.h>
int main()
}
int x;
clrscr();
cout << "Number:=";
cin >> x;
 if ((x % 5)==0)        // if (!(x % 5))
cout << " --- Number accept division by 5 without remainder ---";
else
cout << " --- Number don't accept division by 5 without remainder ---";

cout << "\n\nHit any key to continue";
getch()
return 0;
}
```

Q) Write a program to solve the following equations

$Y=x+1-y$ when $x>=20$

$Y=(x-1)*y$ when $x<20$

```cpp
#include <iostream.h>
#include <conio.h>
int main()
{
int x,y;
clrscr();
cout << "x:=";
cin >> x;
cout << "y:=";
cin >> y;

 if (x >=20)
  {
  y=x+1-y;
  cout << " y:="<<y;
  }
 else
 {
   y=(x-1)*y;
  cout << " y:="<<y;

  }
cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Note: It is better to write the following program rather the previous
one

```cpp
#include <iostream.h>
#include <conio.h>
int main()
{
int x,y;
clrscr();
cout << "x:=";
cin >> x;
cout << "y:=";
cin >> y;

 if (x >=20)
  y=x+1-y;
else
   y=(x-1)*y;

cout << " y:="<<y;//because in both cases, y is the output
cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

If/else if/ else statement

The (if/else if) statement is a chain of if statements. They perform their tests, one after the other, until one of them is found to be true.

This construction is like a chain of (if/else) statements. The else part of one statement is linked to the if part of another. When put together this way, the chain of if/else becomes one long statement.

```cpp
if (condition1)
{
    // code to execute if condition1 is true
}
else if (condition2)
{
//code to execute if condition1=false but condition2=true
}
Else
{
//code to execute if both condition1 and condition2=false
}
```

- Or it can be written as:

```cpp
if (condition1)
{
    // code to execute if condition1 is true
}
Else
{
 if (condition2)
{
//code to execute if condition1=false but condition2=true
}
Else
{
//code to execute if both condition1 and condition2=false
}
}
```

Here's slightly more complex example:

```cpp
#include <iostream.h>
int main()
{
    int b;
    cout << "Enter a value:";
    cin >> b;
    if (b < 0)
        cout << "The value is negative";
    else if (b == 0)
        cout << "The value is zero";
    else
        cout << "The value is positive";
    return 0;
}
```

Q) Write a program to solve the following equations

y=x+1-y when x>20

y=(x-1)*y   when x<20

y=x   when x=20

```cpp
#include <iostream.h>
#include <conio.h>
int main()
{
int x,y;
clrscr();
cout << "x:=";
cin >> x;
cout << "y:=";
cin >> y;

 if (x >20)
  y=x+1-y;
 else if (x<20)
   y=(x-1)*y;
  else
   y=x;

  cout << " y:="<<y;
  cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to solve the following equations

a=a/10   when a>b

b=a/3   when a<b

a=a*b   when a=b

```cpp
#include <iostream.h>
#include <conio.h>
int main()
{
float a,b;
clrscr();
cout << "a:=";
cin >> a;
cout << "b:=";
cin >> b;

 if (a >b)
 {
  a/=10.0;
  cout << " a:="<<a;
 }
 else if (a<b)
  {
   b=a/3.0;
   cout << " b:="<<b;
  }
  else
  {
   a=a*b;
   cout << " a:="<<a;
  }
  cout << "\n\nHit any key to continue";
getch();
return 0;
}
```