### Decision making in C++

Decision making is about deciding the order of execution of statements based on certain conditions or repeat a group of statements until certain specified conditions are met. C++ handles decision-making by supporting the following statements,

* *if* statement
* *switch* statement
* conditional operator statement
* *goto* statement

#### Decision making with if statement

The *if* statement may be implemented in different forms depending on the complexity of conditions to be tested. The different forms are,

1. Simple *if* statement
2. *If....else* statement
3. Nested *if....else*statement
4. *else if* statement

#### Simple if statement

The general form of a simple *if* statement is,

if( expression )

{

statement-inside;

}

statement-outside;

If the *expression* is true, then 'statement-inside' it will be executed, otherwise 'statement-inside' is skipped and only 'statement-outside' is executed.

**Example :**

#include< iostream.h>

int main( )

{

int x,y;

x=15;

y=13;

if (x > y )

{

cout << "x is greater than y";

}

}

Output : x is greater than y

#### if...else statement

The general form of a simple *if...else* statement is,

if( expression )

{

statement-block1;

}

else

{

statement-block2;

}

If the 'expression' is true, the 'statement-block1' is executed, else 'statement-block1' is skipped and 'statement-block2' is executed.

**Example :**

void main( )

{

int x,y;

x=15;

y=18;

if (x > y )

{

cout << "x is greater than y";

}

else

{

cout << "y is greater than x";

}

}

Output : y is greater than x

#### Nested if....else statement

The general form of a nested *if...else* statement is,

if( expression )

{

if( expression1 )

{

statement-block1;

}

else

{

statement-block 2;

}

}

else

{

statement-block 3;

}

if 'expression' is false the 'statement-block3' will be executed, otherwise it continues to perform the test for 'expression 1' . If the 'expression 1' is true the 'statement-block1' is executed otherwise 'statement-block2' is executed.

**Example :**

void main( )

{

int a,b,c;

clrscr();

cout << "enter 3 number";

cin >> a >> b >> c;

if(a > b)

{

if( a > c)

{

cout << "a is greatest";

}

else

{

cout << "c is greatest";

}

}

else

{

if( b> c)

{

cout << "b is greatest";

}

else

{

printf("c is greatest");

}

}

getch();

}

#### else-if ladder

The general form of else-if ladder is,

if(expression 1)

{

statement-block1;

}

else if(expression 2)

{

statement-block2;

}

else if(expression 3 )

{

statement-block3;

}

else

default-statement;

The expression is tested from the top(of the ladder) downwards. As soon as the true condition is found, the statement associated with it is executed.

**Example :**

void main( )

{

int a;

cout << "enter a number";

cin >> a;

if( a%5==0 && a%8==0)

{

cout << "divisible by both 5 and 8";

}

else if( a%8==0 )

{

cout << "divisible by 8";

}

else if(a%5==0)

{

cout << "divisible by 5";

}

else

{

cout << "divisible by none";

}

getch();

}

#### Points to Remember

1. In *if* statement, a single statement can be included without enclosing it into curly braces { }
2. int a = 5;
3. if(a > 4)
4. cout << "success";

No curly braces are required in the above case, but if we have more than one statement inside *if* condition, then we must enclose them inside curly braces.

1. == must be used for comparison in the expression of *if* condition, if you use = the expression will always return true, because it performs assignment not comparison.
2. Other than **0(zero)**, all other values are considered as true.
3. if(27)
4. cout << "hello";

In above example, hello will be printed.