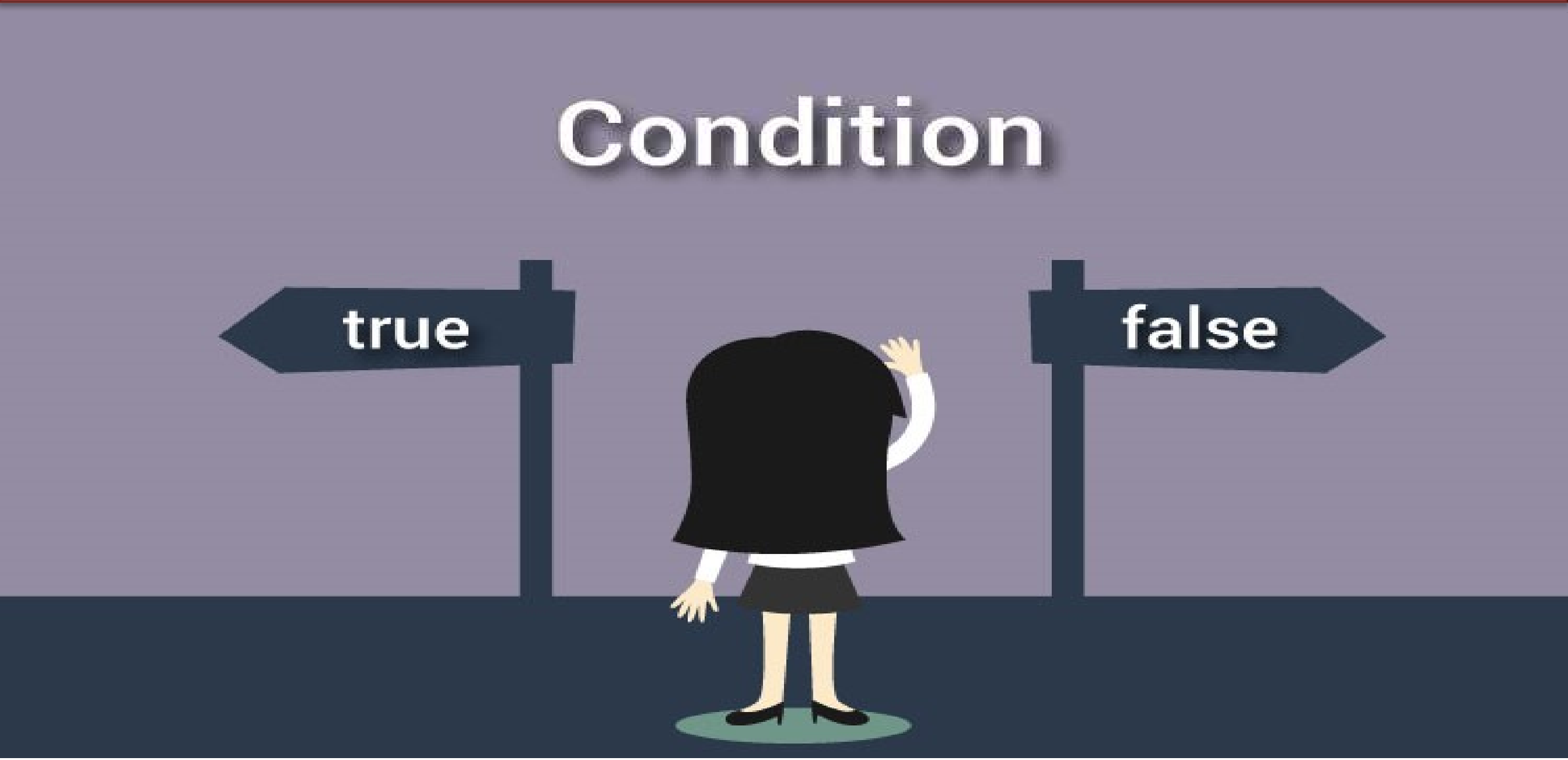
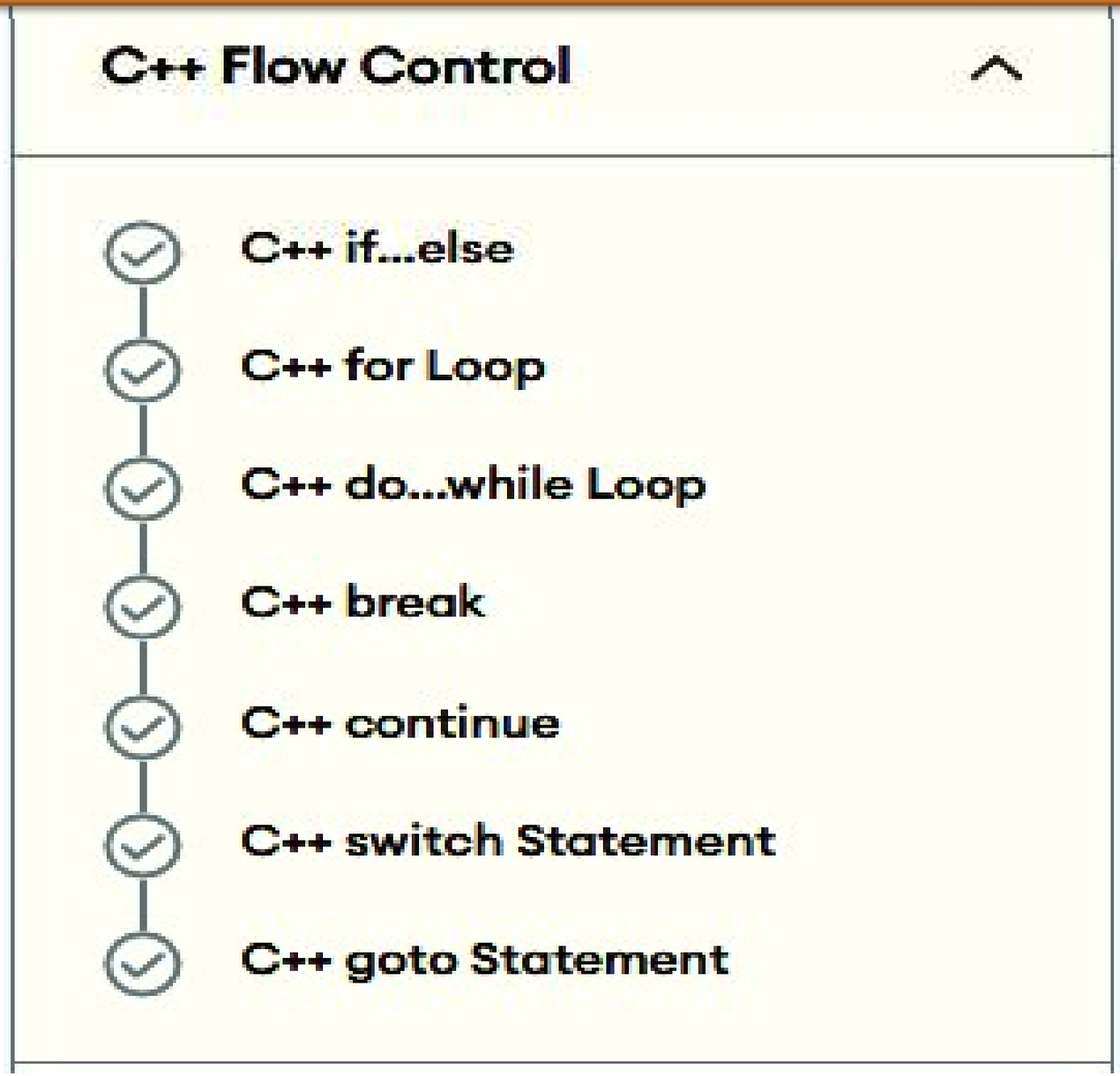
FLOW CONTROL

Control structure



TOPICS TO BE COVERED



CONTROL STRUCTURE

Control structure

C control structure are divided into two parts:

- **Decisions making control structure If**
  - ✓ if
  - ✓ If else
  - ✓ Else If
  - ✓ Nested If
  - ✓ Switch(case control structure
- **Loop control structure**
  - ✓ For loop
  - ✓ While loop
  - ✓ Do while loop
- **Continue, break keywords**

If control structure

C++ if...else

There are three forms of if...else statements in C++.

1. if statement
2. if...else statement
3. if...else if...else statement

If control structure

- The if statement evaluates the test expression inside parenthesis.
- If test expression is evaluated to true (nonzero), statements inside the body of if is executed.
- If test expression is evaluated to false (0), statements inside the body of if is skipped.
- Syntax:-

```
if (Condition)
 {
        codes inside the body of if
 }
```

C++ if Statement

Condition is true

```
int number = 5;

if (number > 0) {
    // code
}

// code after if
```

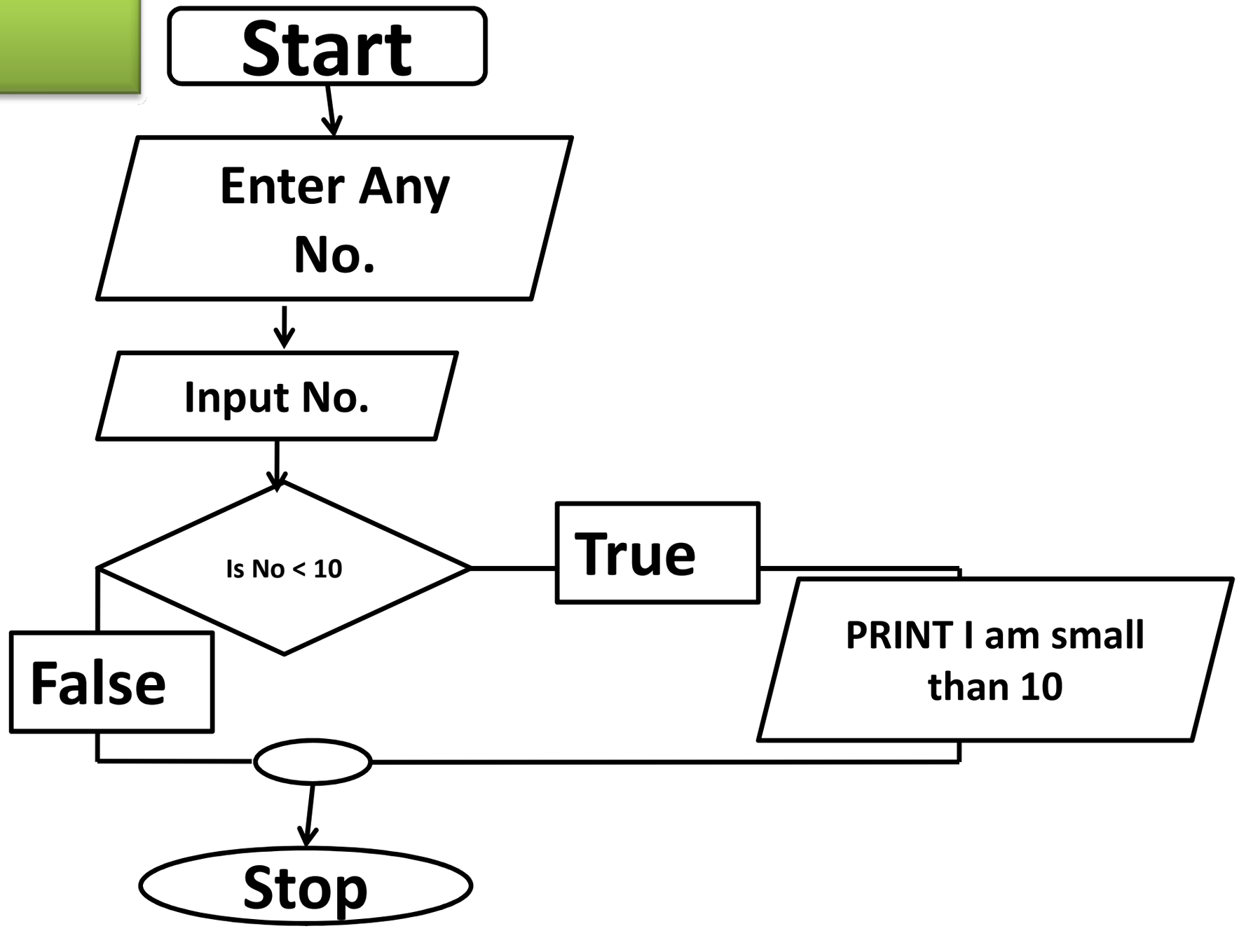
Condition is false

```
int number = 5;

if (number < 0) {
    // code
}

// code after if
```

Flowchart


Start
Enter Any No.
Input No.
Is No < 10
True
PRINT I am small than 10
False
Stop

if... statement

```
void main()
{
int a;
cout<<"enter your age:";
cin>>a;
if(a>18)
{
cout<<"welcome:\n"<<a;
}
}
```

if... statement

```
void main()
{
int a;
cout<<"enter your age to vote:"<<endl;
cin>>a;
if (a>18)
{
cout<<"you  are eligible to vote:"<<endl;
}
{
cout<<“Thank you "<<endl;
 } }
```

C++ if...else

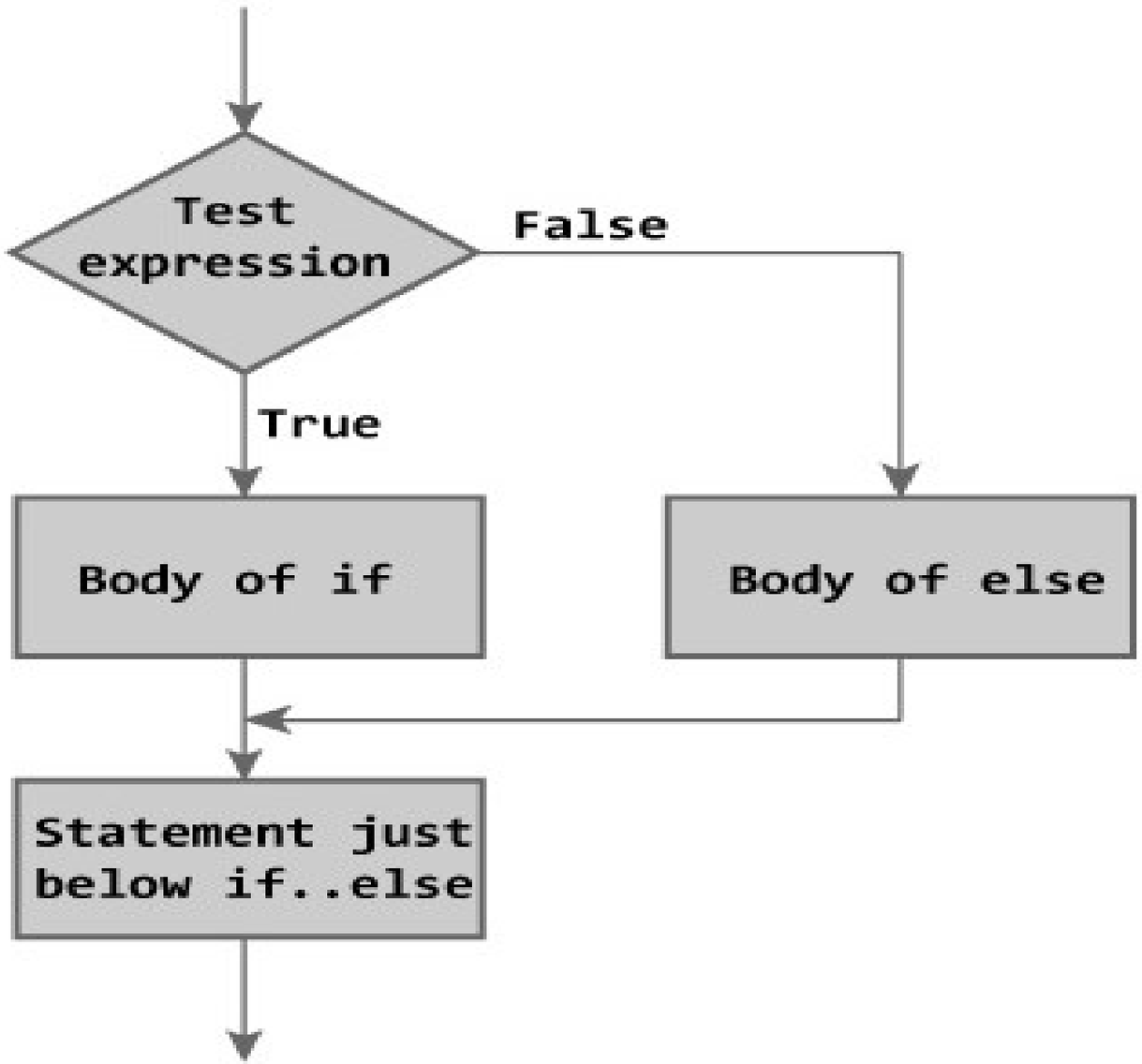
If else statement :

if else statement will perform the task even though condition become false.

Syntax:-

```
if (Condition)
    {
    codes inside the body of if
    }
else
    {
    codes inside the body of else
    }
```

Flowchart of if...else statement



if...else Statement

```
Void main()
{
int a;
cout << "Enter an integer: ";
cin >> a;
if (a >= 0)
{
cout << "You entered a positive integer: " << a << endl;
}
else
{
cout << "You entered a negative integer: " << a << endl;
}
}
```

if… else statement

Program to display a a if user enters negative a If user enters positive a, that a won't be displayed

```
void main()
{
clrscr();
int a;
cout<<"enter your age:\n";
cin>>a;
if(a>18)
{
cout<<"welcome\n"<<a<<endl;
}
else
{
cout<<"not allowed\n";
}
getch();
}
```

if...else statement - to check even or odd

```
#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
int a;
cout<<"enter an integer:";
cin>>a;
if(a%2==0)
{
cout<<"is an even
integer"<<a<<endl;
}
else
{
cout<<"is an odd
integer"<<a<<endl;
}
getch(); }
```

if else Statement without bracket

```
void main()
{
int a;
cout << "Enter an integer: ";
cin >> a;
if (a > 0)
cout<<"You entered a positive integer."<<a<< endl;
else
cout << " This is not a positive integer .";
}
```

C++ if...else

Condition is true

```
int number = 5;

if (number > 0) {
    // code
}
else {
    // code
}

// code after if...else
```

Condition is false

```
int number = 5;

if (number < 0) {
    // code
}
else {
    // code
}

// code after if...else
```

C++ if...else if.....else

if...else...else if statement

1st Condition is true

```
int number = 2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

2nd Condition is true

```
int number = 0;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

All Conditions are false

```
int number = -2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

C++ if...else if...else

```
main()
{
int a;
cout << "Enter an integer: ";
cin >> a;
if (a > 0)
{
cout << "You entered a positive integer: " << a << endl;
}
else if (a < 0)
{ cout << "You entered a negative integer: " << a << endl;
}
else
{
cout << “ You entered 0 " << endl;  } }
```

Program to relate two integers using =, > or <

```
void main()
{
clrscr();
int a,b;
cout<<"enter two integer:";
cin>>a>>b;
if(a==b)
{
cout<<a<<"="<<b;
}
else if(a>b)
{
cout<<"result:"<<a<<">"<<b<<end
l;
}
else
{
cout<<"result:"<<a<<"<"<<b<<end
l;
}
getch();
}
```

PROGRAM TO FIND THE BIGGEST OF THREE

```
void main()
{
clrscr();
int a,b,c;
cout<<"enter the value for a:\n";
cin>>a;
cout<<"enter the value for b:\n";
cin>>b;
cout<<"enter the value for c:\n";
cin>>c;
if((a>b)&&(a>c))
{
cout<<"\n the big one is a:  "<<a;
}
else if(b>c)
{
cout<<"\n the big one is b:  "<<b;
}
else
{
cout<<"\n the big one is c:  "<<c;
} }
```

Nested if statement

C++ Nested if...else

```
if (condition1)
    {
        // statements
        if (condition2)
            {
                // statements
            }
    }
```

Syntax:

```
if (condition1)
    {
        if(condition2)
            {
              do this
             }
            else
             {
             do this
             }
    }
else
    {
    do this
    }
```

Nested If: putting of if else statement within the body of the if statement or the body of an else statement this is called nested if.

Login & Password

```
void main()
{
clrscr();
char username;
int password;
cout<<"username:";
cin>>username;
if(username=='a')
    {
      cout<<"password:";
      cin>>password;
            if(password==12345)
                {
                cout<<"login successful";
                }
     }
     else
     {
cout<<"user name does not exists, try again.";
} }
```

Nested if

```
int main()
{
int a;
cout << "Enter an integer: ";
cin >> a;
    if (a != 0)
        {
           if ((a % 2) == 0)
           {
           cout << "The number is even.”;
           }
            else
           {
           cout << "The number is odd." <<
    endl;
           }
       }
else
{
cout <<“Number is 0 & it is neither even nor odd";
}
}
```

Body of if...else With Only One Statement

If the body of if...else has only one statement, you can omit { } in the program.

```
int number = 5;
if (number > 0)
{
cout << "The number is positive."<< endl;
}
else
{
cout << "The number is negative." << endl;
}
```

Body of if...else With Only One Statement

```
int number = 5;
if (number > 0)
cout << "The number is positive." << endl;
else
cout << "The number is negative." << endl;
```

Ternary operator

Conditional or Ternary Operator (?:) in C/C++

Syntax

condition ? value_if_true : value_if_false

Program to Find Largest Among Two Numbers Using Ternary Operator

```
void main()
{
int a = 5, b = 10, c;  // variable declaration
c = (a > b) ? a : b;  // Largest among n1 and n2
cout<<"Largest number between ”<<a << “& ”<< b <<“is “ <<c;
}
```

Switch statement

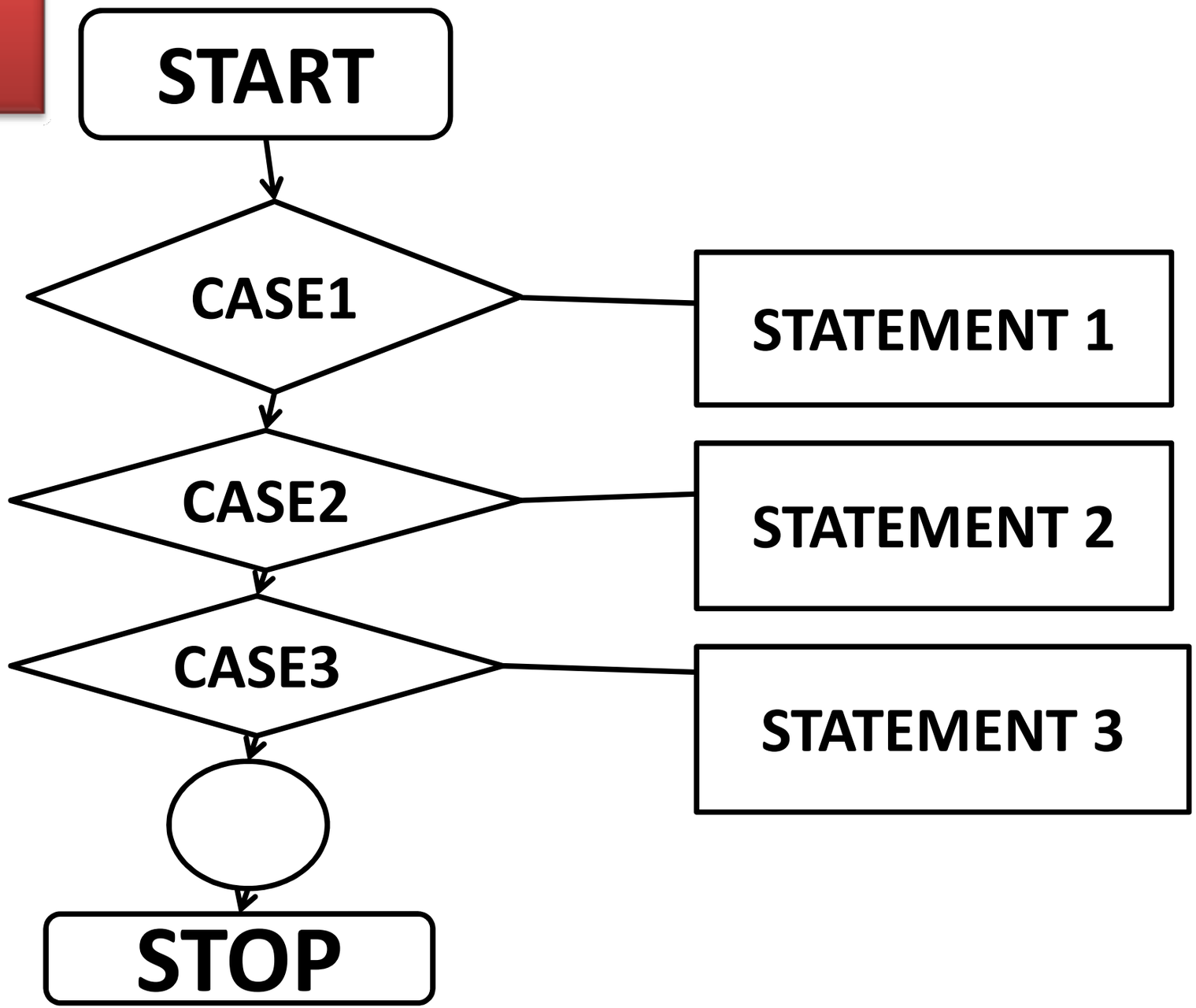
Switch statement

- **The control statement that allow us to make a decision from the choice given by user is called a switch Statement.**

Syntax:

```
Switch(integer expressions)
{
    Case 1: do this;
    Break;
    Case 2: do this;
    Break;
    Case 3: do this;
    Break;
    default: do this;
}
```

Switch case:


START
CASE1
STATEMENT 1
CASE2
STATEMENT 2
CASE3
STATEMENT 3
STOP

Switch Statement for (+, -, x, ÷)

```
void main()
{
clrscr();
int a,b,c,i;
cout<<"enter your choice between 1 to 4<<endl;
cin>>i;
switch(i)
{
case 1:
cout<<"you are at case 1 - Addition\n"<<endl;
cout<<"enter two number\n"<<endl;
cin>>a>>b;
c=a+b;
cout<<"addition is:\n"<<c<<endl;
break;
```

```
case 2:
cout<<"you are at case 2 - Subraction\n"<<endl;
cout<<"enter two number\n"<<endl;
cin>>a>>b;
c=a-b;
cout<<"subtraction is:\n"<<c<<endl;
break;
```

```
case 3:
cout<<"you are at case 3 - Multiplication\n"<<endl;
cout<<"enter two number\n"<<endl;
cin>>a>>b;
c=a*b;
cout<<"mutiplication is:\n"<<c<<endl;
break;
```

```
case 4:
cout<<"you are at case 4-division\n"<<endl;
cout<<"enter two number\n"<<endl;
cin>>a>>b;
c=a/b;
cout<<"division is:\n"<<c;
break;

        default:
        cout<<"invalid selection enter a valid choice"<<endl;
        }
        getch();
        }
```

Switch Statement for (+, -, x, ÷)

```
void main()
{
clrscr();
int a,b,c,i;
cout<<"enter your choice between 1 to 4\n
Enter 1 for addition\n
Enter 2 for subtraction\n
Enter 3 for multiplication\n
Enter 4 for division\n"<<endl;
cin>>i;
switch(i)
{
case 1:
cout<<"you are at case 1 - Addition\n"<<endl;
cout<<"enter two number\n"<<endl;
cin>>a>>b;
c=a+b;
cout<<"addition is:\n"<<c<<endl;
break;
```

Goto statement

Goto statement

```
void main()
{
cout<<"go to statement is"<<endl;
goto x;
y:
cout<<"uncondition jumb statement"<<endl;
goto z;
x:
cout<<"is a"<<endl;
goto y;
z:
cout<<“It make program hard to understand"<<endl;
getch(); }
```

Goto statement

- Control may be transferred to anywhere within the current function .
- The target statement must be labeled , and a colon must follow the label.
- **NOTE :**
- use of goto statement is highly discouraged in any programming language because
- it makes difficult to trace the control flow of a program hard to understand and hard to modify .


Thank You!