Lecture 6: Control Structures

A control structure is a block of programming that analyzes variables and chooses a direction in which to go based on given parameters. Flow of control through any given function is implemented with three basic types of control structures:

- ❖ Sequential: default mode. ...
- ❖ Selection: used for decisions, branching -- choosing between 2 or more alternative paths. ...
- ❖ Repetition: used for looping, i.e. repeating a piece of code multiple times in a row.

4.1 If else Statement

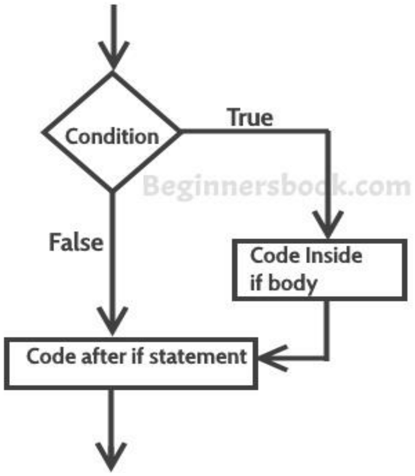
Sometimes we need to execute a block of statements only when a particular condition is met or not met. This is called **decision making**, as we are executing a certain code after making a decision in the program logic. For decision making in C++, we have four types of control statements (or control structures), which are as follows:

a) if statement
b) nested if statement
c) if-else statement
d) if-else-if statement

a) if statement If statement consists a condition, followed by statement or a set of statements as shown below:

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

Flow diagram of If statement



```
      #include <iostream>
      using namespace std;
     int main(){
  int num=70;
  if( num < 100 ){
     /* This cout statement will only execute,
      * if the above condition is true
      */
     cout<<"number is less than 100";
  }

  if(num > 100){
     /* This cout statement will only execute,
      * if the above condition is true
      */
     cout<<"number is greater than 100";
  }
  return 0;
}
```

Output:

```
number is less than 100
```

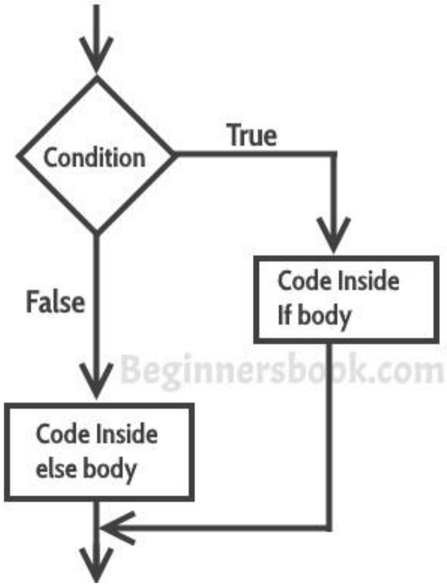
b) nested if statement When there is an if statement inside another if statement then it is called the **nested if statement**. The structure of nested if looks like this:

```
if(condition_1) {
   Statement1(s);

   if(condition_2) {
      Statement2(s);
   }
}
```

c) if-else statement :

Flow diagram of if-else



```
#include <iostream>
using namespace std;
int main(){
   int num=90;
   /* Nested if statement. An if statement
    * inside another if body
    */
   if( num < 100 ){
      cout<<"number is less than 100"<<endl;
      if(num > 50){
         cout<<"number is greater than 50";
      }
   }
   return 0;
}
```

Output:

```
number is less than 100
number is greater than 50
```

d)if-else-if Statement

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:

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

```
else {
   /* if none of the condition is true
    * then these statements gets executed
    */
     statement(s);}
  #include <iostream>
 using namespace std;
      int main(){
   int num;
   cout<<"Enter an integer number between 1 & 99999: ";
   cin>>num;
   if(num <100 && num>=1) {
      cout<<"Its a two digit number";
   }
   else if(num <1000 && num>=100) {
      cout<<"Its a three digit number";
   }
   else if(num <10000 && num>=1000) {
      cout<<"Its a four digit number";
   }
   else if(num <100000 && num>=10000) {
      cout<<"Its a five digit number";
   }
   else {
      cout<<"number is not between 1 & 99999";
   }
   return 0;
}
```

Output:

```
Enter an integer number between 1 & 99999: 8976
Its a four digit number
```

4.2 Switch Case statement

Switch case statement is used when we have multiple conditions and we need to perform different action based on the condition. In such case either we can use lengthy if..else-if statement or switch case. The problem with lengthy if..else-if is that it becomes complex when we have several conditions. The switch case is a clean and efficient method of handling such scenarios.

The **syntax of Switch case** statement:

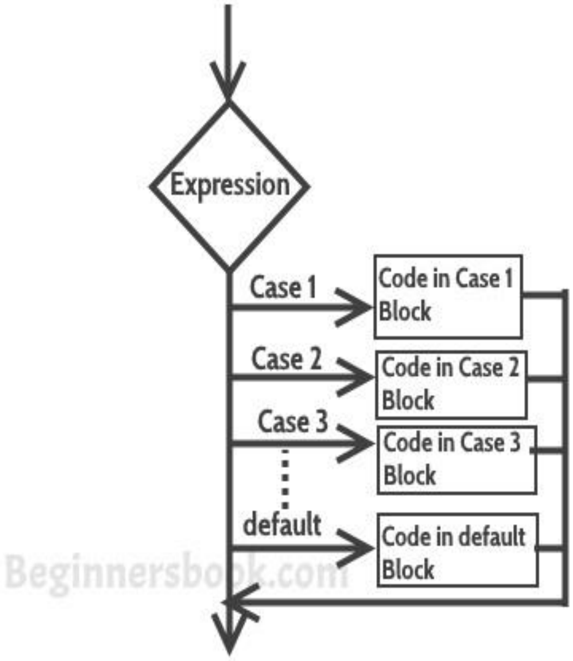
```
switch (variable or an integer expression)
{
     case constant:
     //C++ code
     ;break;
     case constant:
```

```
//C++ code
; break;
default:
//C++ code
;}
```

```
#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;
}
```

Output:

```
Case2
```



```
  #include <iostream>
 using namespace std;
int main(){
   char ch='b';
   switch(ch) {
      case 'd': cout<<"Case1 ";
      break;
      case 'b': cout<<"Case2 ";
      break;
      case 'x': cout<<"Case3 ";
      break;
      case 'y': cout<<"Case4 ";
      break;
      default: cout<<"Default ";
   }
   return 0;
}
```