NOI 2020 C++ Basics Course Content

Note 9

Conditions

Now that we know how to use variables let's move onto decision making.
The logical operators mentioned in the previous section can be used to generate true or false as the answer. These can be used to make decisions on what to happen in the program.

- If statement
  Syntax :
  if(<boolean condition>){
          <code to run if the given **condition** is **true**>
  }

Eg:

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

int main() {
      int myVariable = 20;
      int myVariable1 = 40;
      if(myVariable > myVariable1){
            cout << myVariable;
      }
      return 0;
}
```

```
Here the code inside if block is not run because the given condition
is false
myVariable > myVariable1 is false
Output : won't print anything
```

- If else statement

  Syntax :

  if(<boolean condition>){

  <code to run if the given **condition** is **true**>

  } else{

  <code to run if the given **condition** is **false**>

  }

  Eg:

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

int main() {
      int myVariable = 20;
      int myVariable1 = 40;
      if(myVariable > myVariable1){
            cout << myVariable;
      }else{
            Cout << myVariable1;
      }
      return 0;
}
```

Here the given condition is false

Therefore the code block of if statement will not run, the code block inside else statement will run.

Output : 40

- else if statement
  This statement can be used to implement multiple conditions. Works the same way if else statement worked.

  Syntax :

  ```
  if( <boolean condition1> ){
          <code to run if the given condition1 is true>
  } else if(  <boolean condition2> ){
          <code to run if the given condition2 is true>

  }else{
          <code to run if all the above given conditions are false>
  }
  ```

Extra resources:

- If else statement
  Video:
  - https://www.youtube.com/watch?v=jK83lln_T1k&t=164s

  Reading:
  - https://www.tutorialspoint.com/cplusplus/cpp_decision_making.htm
  - https://www.w3schools.com/cpp/cpp_conditions.asp
  - https://www.w3schools.com/cpp/cpp_conditions_else.asp
  - https://www.w3schools.com/cpp/cpp_conditions_elseif.asp

- Ternary operator

  This is the shorter version of th if else statement. This is often used to replace small if else conditions.

  Syntax:

  (<condition>)? <statement to run if true>: <statement to run if false>;

  Syntax to include multiple statements to run for a state

  (<condition>)? (<statement1>,< statement2>..): (<statement3>,<statement4>,...)

  Here statement1 and 2 will be run if the condition is true and statement 3 and 4 if the condition is false.

  Eg:

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

    int main() {
          int myVariable = 20;
          int myVariable1 = 40;

    int result = (myVariable > myVariable1) ?
    myVariable:myVariable1;

    cout << result;
          return 0;
    }
```

```
Output : 40
```

Extra resources:

- Ternary operator

  Video:

  - https://www.youtube.com/watch?v=WSy25XB7Eak

  Reading:

  - https://www.w3schools.com/cpp/cpp_conditions_shorthand.asp

- Switch statement

  Switch is a good alternative to using multiple if else statements, if at each condition you are checking the value of the same variable.

  Syntax :

  ```
  switch (<variable name>){
          case <value1>:
                  <code to run if value of variable equals value1>
                  break;
          case <value2>:
                  <code to run if value of variable equals value2>
                  break;
          case <value3>:
                  <code to run if value of variable equals value3>
                  break;
          case <value4>:
                  <code to run if value of variable equals value4>
                  break;
          default:
                  <code to run if value of variable equals none mentioned above>
  }
  ```

Eg:

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

int main() {
    int myVariable = 20;
    int myVariable1 = 40;

    int result = (myVariable > myVariable1) ?
    myVariable:myVariable1;

    switch(result):
        case 20:
            cout<<”myVariable is larger”;
            break;
        case 40:
            cout<<”myVariable1 is larger”           ;
            break;
        default:
            cout<<”myVariable and myVariable1 are equal”;

    return 0;
}

Output : myVariable1 is larger
```

Extra resources:

- Switch

  Video:
  - https://www.youtube.com/watch?v=GSja2w-HN20
  - https://www.youtube.com/watch?v=sQkgGd7PEfM

  Reading:
  - https://www.tutorialspoint.com/cplusplus/cpp_switch_statement.htm
  - https://www.w3schools.com/cpp/cpp_switch.asp