EXPERIMENT NO 4

Nested If Else and Switch Statement

Objectives:

- Learn how to use nested if/else selection statements and use it to write and execute a simple program.
- Learn the structure of the switch statement.
- Learn how to convert between if/else and switch statement.

Equipment required:

- Dev-C++/Eclipse/Visual Studio installed in PC/Windows

DISCUSSION

1. Pre-Lab

Nested If else

In C++ we can use if statement in another else block or else if block and we can also include if block in another if block

Syntax

```
if(boolean_expression1)   // Executes when the Boolean expression 1 is true
{

if(boolean_expression2)   // Executes when the Boolean expression 2 is true
{

if(boolean_expression3)   // Executes when the Boolean expression 3 is true
{
}

else   // Executes when the Boolean expression 3 is false.
{
}

}
}
```

Example

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

int main()
{
    int age = 87;

    if(age>60){

        if(age>100){
            cout << "why are you stil alive?"
        }

    }else{
        cout << "you are young, get a job" << endl;
    }

    return 0;
```

We can nest else if…else in the similar way as you have nested if statement in the above example. Similarly, we can nest any conditional statement in any if/else or else if block.

Switch Statement

Switch is a branching statement used to perform action based on available choices, instead of making decisions based on conditions. Using switch...case you can write more clean and optimal code than if...else statement. Switch only works with integer, character, and enumeration constants.

Syntax

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

This is how it works:

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.

- When C++ reaches a break keyword, it breaks out of the switch block.
- This will stop the execution of more code and case testing inside the block.
- When a match is found, and the job is done, it's time for a break. There is no need for more testing.
- A break can save a lot of execution time because it "ignores" the execution of all the rest of the code in the switch block.
- The default keyword specifies some code to run if there is no case match

Example

```
int day = 4;
switch (day) {
  case 6:
    cout << "Today is Saturday";
    break;
  case 7:
    cout << "Today is Sunday";
    break;
  default:
    cout << "Looking forward to the Weekend";
}
// Outputs "Looking forward to the Weekend"
```

2. Post-Lab (Lab Tasks)

Note: Use switch statement only (if you think the problem can't be solved using switch, write the reason and solve it using if else).

1. Write a C++ program that print the total number of days in a month number entered by user
2. Write a C++ program to check whether an alphabet is vowel or consonant
3. Make a simple calculator. The program should ask the user to input two operands and an operation type.
4. Write a program that reads an alphabet from user and then find if it is capital letter or small letter.
5. Write a program to read four integer numbers then find and print the second maximum number among these numbers.
6. Enter a 4-digit integer number from user. Check whether 100th place digit of the entered number is even or add.
7. Write a program to check whether a triangle is valid or not, when the three angles of the triangle are entered by the user. A triangle is valid if the sum of all the three angles is equal to 180 degrees.

END