Lab 04

Objectives

After performing this lab, students will be able to

- Understand nested if (or if-else) statement
- Understand and use switch-case statement

Nested if statement

Nested if means you can put an **if** statement (including any **else if – else**) inside another **if** or **else if** statement(s). Its general syntax is:

```
if( boolean_expression 1) {
    // Executes when the boolean expression 1 is true
    if(boolean_expression 2) {
        // Executes when the boolean expression 2 is true
    }
}
```

You can use **else if – else** in the similar way as **if** is nested in the above-given syntax.

Example:

```
// C++ program to find if an integer is even or odd or neither (0)
// using nested if statements
#include <iostream>
using namespace std;

int main() {
    int num;
    cout << "Enter an integer: ";
    cin >> num;

    // outer if condition
    if (num != 0) {
        // inner if condition
        if ((num % 2) == 0) {
            cout << "The number is even." << endl;
        }
        // inner else condition
        else {
```

```
      cout << "The number is odd." << endl;
    }
  }
  // outer else condition
  else {
    cout << "The number is 0 and it is neither even nor odd." << endl;
  }
  cout << "This line is always printed." << endl;

}
```

Output: 1

Enter an integer: 34
The number is even.
This line is always printed.

Output: 2

Enter an integer: 35
The number is odd.

This line is always printed.

Output: 3

Enter an integer: 0
The number is 0 and it is neither even nor odd.
This line is always printed.

In the above example,

- We take an integer as an input from the user and store it in the variable num.
- We then use an if...else statement to check whether num is not equal to 0.
  - If true, then the inner if...else statement is executed.
  - If false, the code inside the outer else condition is executed, which prints "The number is 0 and neither even nor odd."
- The inner if...else statement checks whether the input number is divisible by 2.
  - If true, then we print a statement saying that the number is even.
  - If false, we print that the number is odd.

Notice that 0 is also divisible by 2, but it is actually not an even number. This is why we first make sure that the input number is not 0 in the outer if condition.

The switch statement

A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case. Its general syntax is:

```
switch(expression) {
   case constant-expression  :
      statement(s);
      break; //optional
   case constant-expression  :
      statement(s);
      break; //optional

   // you can have any number of case statements.
   default : //Optional
      statement(s);
}
```

The following rules apply to a switch statement

- The expression used in a switch statement must have an integral type (int, char etc.).
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon "**:**".
- The constant-expression for a case must be the same data type as the variable in the switch.
- When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.
- When a **break** statement is reached, the switch terminates, and the flow of control jumps to the next line following the **switch** statement.
- Not every case needs to contain a **break**. If no **break** appears, the flow of control will fall through to subsequent cases until a **break** is reached.
- A switch statement can have an optional **default** case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true (it is similar to **else** statement in **if-else**). No break is needed in the default case.

Example:

```cpp
int a;
cout << "Enter any number: ";
cin >> a;

switch (a) {
    case 100:
        cout << "Value is 100";
        break;
    case 200:
        cout << "Value is 200";
        break;
    case 300:
        cout << "Value is 300";
        break;
    default:
        cout << "Value is other than 100 or 200 or 300";
}
```

Output 1:

```
Enter any number: 300
Value is 300
```

Output 2:

```
Enter any number: 100
Value is 100
```

Output 3:

```
Enter any number: 56
Value is other than 100 or 200 or 300
```

The use of **break** is important. Otherwise, when one case becomes true, all the subsequent case (cases after that case) will also be executed regardless of whether their condition is true or false.

Example 2 (where **break** statement is NOT used):

```cpp
int a;
cout << "Enter any number: ";
cin >> a;

switch (a) {
    case 100:
        cout << "Value is 100" <<endl;
    case 200:
        cout << "Value is 200" <<endl;
    case 300:
        cout << "Value is 300" <<endl;
    default:
        cout << "Value is other than 100 or 200 or 300";
}
```

Output 1:

```
Enter any number: 100
Value is 100
Value is 200
Value is 300
Value is other than 100 or 200 or 300
```

Output 2:

```
Enter any number: 200
Value is 200
Value is 300
Value is other than 100 or 200 or 300
```

Here, you can see that when a particular case becomes true, then all the cases after it (and the *default* case) execute even if their condition is not true. Therefore, it is necessary to use the *break* statement to avoid such behavior of programs.

Exercises:

1. Write a C++ program that should ask the user if he/she has the student ID card or not. If the user enters 'n' or 'N', the program should display a message like "You are not allowed to enter". If the user says 'y' or 'Y', then your program should ask him/her what is his/her department, and then display a respective message as follows:
   a. If the department is CS, then display a message "Go to Block 1"
   b. If the department is EE, then display a message "Go to Block 3"
   c. If the department is BBA, then display a message "Go to Block 2"
   d. If the department is EDU, then display a message "Go to Knowledge Center"
   See some sample output below:

Output 1

```
Do you have student ID card? (Y/N) n
You are not allowed to enter
```

Output 2

```
Do you have student ID card? (Y/N) y
CS
EE
BBA
EDU
What is your department name? CS
Go to Block 1
```

Output 3

```
Do you have student ID card? (Y/N) Y
CS
EE
BBA
EDU
What is your department name? EDU
Go to Knowledge Center
```

2. Using the switch statement
   a. Write a program that asks the user to input two numbers and an operator (+, -, *, or /), and then performs the operation (addition, subtraction, multiplication, or division) on the two entered numbers according to the operator input by the user. If the entered operator is not from these four operators, the program should display an error message saying that the entered operator is invalid.
   b. Write a program that should ask the user to input a month number (from 1 to 12) and then display the respective month name in words (like January for 1, February for 2 and so on). If the user inputs value other than numbers from 1 to 12, the program should display a message saying that it is an invalid month number.
   c. Write a program that ask the user to input a month (from 1 to 12) and then display which season currently it is. The season should be displayed as per the following rules
      i. If the month number is 12 or 1 or 2, then it is the Winter season
      ii. If the month number is 3 or 4, then it is the Spring season
      iii. If the month number is 5 or 6 or 7 or 8 or 9, then it is the Summer season
      iv. If the month number is 10 or 11, then it is the Autumn season
   d. Write a program that should ask the user to input an alphabet character (small or capital) and tell if it is a vowel or a consonant.
   e. Write a program that should ask the user to input an integer number and print whether it is even or odd.