

Lab 6 - Computer Science 1 (CMPS 1044) Switch Lab

Objective: Demonstrate use of the switch statement in C++

Switch Statement: A selection control mechanism that can sometimes be used in place of if/else if statements. Allows the program to branch.

Syntax:

```
switch (IntegerExpression){
    case ConstantExpression1:
        // Statements to be executed
        // if IntegerExpression == ConstantExpression1
        break; //Causes the program to exit the switch statement
    case ConstantExpression2:
        // Statements to be executed
        // if IntegerExpression == ConstantExpression2
        break;
    default:
        // Statements to be executed
        // if IntegerExpression doesn't match any case
}
```

IMPORTANT

- *IntegerExpression* must be a single variable of any integer data type.
- *ConstantExpression* must be integer literal or integer constant. It cannot be an integer variable or Boolean expression.

1. Open a new project and type in the code below. Test it with all cases.

```
char letter;
cout << "Enter a W, D, or T to indicate transaction \n";
cin >> letter;
switch (letter){
    case 'W':
        cout << "Withdrawal\n";
        break;
    case 'D':
        cout << "Deposit\n";
        break;
    case 'T':
        cout << "Transfer\n";
        break;
    default:
        cout << "Not a valid choice.\n";
}
```

2. Now, modify your code by commenting out the break statement in the 'W' case as shown below.

```
char letter;
cout << "Enter a W, D, or T to indicate transaction \n";
cin >> letter;
switch (letter){
case 'W':
    cout << "Withdrawal\n ";
    //break;
case 'D':
    cout << "Deposit\n ";
    break;
case 'T':
    cout << "Transfer\n ";
    break;
default:
    cout << "Not a valid choice. \n ";
}
```

3. Re-build and run again. Now what is the output you enter 'W'?
-

This demonstrates the "fall-through" feature of `switch` statements. When a matching case is found, all statements following the case `**`: line will be executed until a `break` statement (or the closing `}`) is encountered...even if one or more case `**`: lines are encountered.

4. Sometimes, this is what the programmer wants. Modify your code to look like this:

```
switch (letter){
case 'W':
case 'w':
    cout << "Withdrawal";
    break;
case 'D':
case 'd':
    cout << "Deposit";
    break;
case 'T':
case 't':
    cout << "Transfer";
    break;
default:
    cout << "Not a valid choice.\n";
}
```

Re-build, run, and test with both uppercase and lowercase responses.

5. Now, modify your program so that it will accomplish the same thing as before, but implements the `if/else if` statement. Show your work to the GA when you finish.

6. What is wrong with the following switch statement?

```
int temp;
cout << "Enter the temperature. ";
cin >> temp;
switch (temp){
case (temp < 0) :
    cout << "Temperature is negative.\n";
    break;
case 0:
    cout << "Temperature is zero.\n";
    break;
case (temp > 0) :
    cout << "Temperature is positive.\n";
    break;
}
```

7. Assignment: Rewrite the following program segment using a switch statement instead of the if/else if statement.

```
int selection;
cout << "Which formula do you want to see?\n\n";
cout << "1. Area of a circle\n";
cout << "2. Area of a rectangle\n";
cout << "3. Volume of a cylinder\n";
cout << "4. None of them!\n";
cin >> selection;
if (selection == 1)
    cout << "Pi times radius squared\n";
else if (selection == 2)
    cout << "Length times width\n";
else if (selection == 3)
    cout << "Pi times radius squared times height\n";
else if (selection == 4)
    cout << "Well okay then...Goodbye!\n";
else
    cout << "Not good with numbers, eh?\n";
```