**Chapter 3 Lab**

**Selection Control Structures**

**Lab Objectives**

* Be able to construct boolean expressions to evaluate a given condition
* Be able to compare strings
* Be able to use a flag
* Be able to construct if and if-else-if statements to perform a specific task
* Be able to construct a switch statement
* Be able to format numbers

**Introduction**

Up to this point, all the programs you have had a sequential control structure. This means that all statements are executed in order, one after another. Sometimes we need to let the computer make decisions, based on the data. A selection control structure (or decision structure) allows the computer to select which statement to execute.

In order to have the computer make a decision, it needs to do a comparison. So we will work with writing boolean expressions. Boolean expressions use relational operators and logical operators to create a condition that can be evaluated as true or false.

Once we have a condition, we can conditionally execute statements. This means that there are statements in the program that may or may not be executed, depending on the condition. We can also chain conditional statements together to allow the computer to choose from several courses of action. We will explore this using nested if-else statements as well as a switch statement.

In this lab, we will be editing a pizza ordering program. It stores the selections that the user makes for the pizza. It walks the user through ordering, giving the user choices, which the program then uses to decide how to make the pizza and how much the cost of the pizza will be. The user will also receive a $2.00 discount if his/her name is Mike or Diane.

**Task #1 The if Statement and Setting a Flag**

1. Create a Visual Studio project and add the file **pizzaOrder.cpp**.
2. Compile and run pizzaOrder.cpp. You will be able to make selections, but at this point, you will always get a hand-tossed crust no matter what you select, but you will be able to choose toppings and they will be the only cost so far. You will also notice that the output does not look like money. So we need to edit pizzaOrder.cpp to complete the program so that it works correctly.
3. Construct a simple if statement. The condition should extract the first character of the string entered by the user in response to the new customer question. Since we do not want the comparison to be case sensitive, change it to upper case before you compare it.
4. If the user is a new customer, set discount to true.

**Task #2 The if-else-if Statement**

1. Write an if-else-if statement that lets the computer choose which statements to execute by the user input size (10, 12, 14, or 16). For each option, the cost needs to be set to the appropriate amount.
2. The default else of the above if-else-if statement should print a statement that the user input was not one of the choices, so a 12 inch pizza will be made. It should also set the size to 12 and the cost to 12.99.
3. Compile, debug, and run. You should now be able to get correct output for size and price (it will still have hand-tossed crust, the output won’t look like money, and no discount will be applied yet). Run your program multiple times ordering a 10, 12, 14, 16, and 17 inch pizza.

## Task #3 Switch Statement

1. Write a switch statement that compares the user’s choice with the appropriate characters (make sure that both capital letters and small letters will work).
2. Each case will assign the the appropriate string indicating crust type to the crust variable.
3. The default case will print a statement that the user input was not one of the choices, so a hand-tossed crust will be made.
4. Compile, debug, and run. You should now be able to get crust types other than Hand-tossed. Run your program multiple times to make sure all cases of the switch statement operate correctly.

**Task #4 Using a Flag as a Condition**

1. Write an if statement that uses the flag as the condition. Remember that the flag is a Boolean variable, therefore is true or false. It does not have to be compared to anything.
2. The body of the if statement should contain two statements:
   1. A statement that prints a message indicating that the user is eligible for a $2.00 discount.
   2. A statement that reduces the variable cost by 2.
3. Compile, debug, and run. Test your program using the owners’ names (both capitalized and not) as well as a different name. The discount should be correctly at this time.

**Task #5 Formatting Numbers**

1. Format the output so that monetary output has 2 decimal places. Don’t forget the needed include file.
2. Compile, debug, and run. Your output should be completely correct at this time, and numeric output should look like money.

**Task #6 Testing**

1. Determine which test cases you will need to fully test all of the different branches of the program you have written. Make a chart with hand calculated prices to compare to results when testing.

New customer: 2 possibilities (Y/N)

Pizza size: 5 possibilities (10, 12, 14, 16, other)

Crust: (H, T, D, other)

Toppings: 0-4 extra toppings

1. Test with bad input. What happens when you type a number when it expects a letter? What happens when you type a string when it expects a number? Use [www.cplusplus.com](http://www.cplusplus.com) to figure out how to check for digits and convert to an integer.