Lab 3 – Decision Structures (20 pts)

Lab Objectives

- Be able to construct `boolean` expressions to evaluate a given condition
- Be able to compare `String` objects
- 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 output

Deliverables

This lab has five tasks. When you have all tasks done, run the report in Blackboard. The report is a Blackboard test with short-answer, file-response, multiple-answer, and other types of questions.

In the report, you will provide Java code segments, Java source code files (must have extension *.java*), screenshots of program execution, and your analysis of your results.

Screenshots in your report must show a full screen, so your computer can be identified. Please resize your IDE panels in the way that the required dialog or output is visible along with the source code. Show as much source code as possible.

NOTE:

- Use **Blackboard only** to submit your work; **no email** submission unless your instructor directs it.
- If Blackboard gives you multiple submission attempts (usually three), the **last one** will be evaluated and graded.
- **No late submissions**, **no changes** in your submission after the due date.

Introduction

All the programs in the previous Lab 2 have a sequence structure, i.e., all statements are executed in a sequence, one after another. Sometimes we need to let the computer make decisions, based on the data. A decision structure allows the computer to decide which statement to execute.

To have the computer decide, it needs to do a comparison. So, we will work with writing `boolean` expressions. The `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 creates a pizza ordered to the specifications that the user desires. 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 or her name is Mike or Diane.

To complete Task #5 use materials from the lecture slides or/and from section 3.10 of the textbook.

Task #1 The if Statement, Comparing Strings, and Flags (4 pts)

1. Copy the file *PizzaOrder.java* as directed by your instructor. Correct syntax errors if any, and improve programming style when necessary (indents, newlines, etc.).
2. Compile and run *PizzaOrder.java*. You will be able to make selections, but at this point, you will always get a Hand-tossed pizza at a base cost of $12.99 no matter what you select, but you will be able to choose toppings, and they should add into the price correctly. You will also notice that the output does not look like money. So, we need to edit *PizzaOrder.java* to complete the program so that it works correctly.
3. ~~Construct a simple `if` statement. The condition will compare the `String` input by the user as his or her first name with the first names of the owners, Mike and Diane. Be sure that the comparison is not case sensitive.~~
4. ~~If the user has either first name, set the `discount` flag to `true`. This will not affect the price at this point yet.~~

Task #2 The `if-else-if` Statement (4 pts)

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 pizza size to 12 and the cost to 12.99.
3. Compile, debug, and run. You should now be able to get correct output for the pizza 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 The `switch` Statement (4 pts)~~

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 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 (4 pts)

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:
    a. A statement that prints a message indicating that the user is eligible for a $2.00 discount.
    b. 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 displayed correctly at this time.

Task #5 Formatting Output (4 pts)

"%.2f"

1. Edit the appropriate lines in the `main` method so that any monetary output has 2 decimal places.
2. Compile, debug, and run. Your output should be completely correct at this time, and numeric output should look like money.