# CPSC 1100 – LAB 10

More If/Else Statements and Switch

This lab will deal with using if statements, switch statements, and Boolean operators in java to implement different tasks. The tasks will be assigned from the textbook. The entire lab will be due next week. **PLEASE COMMENT YOUR CODE.** You will have points taken off if you do not comment your code. Keep your code neat.

**Some useful links:**

BlueJ tutorial [www.bluej.org/tutorial/tutorial-201.pdf](http://www.bluej.org/tutorial/tutorial-201.pdf)

Java tutorial home page: <http://docs.oracle.com/javase/tutorial/>

Start here: <http://docs.oracle.com/javase/tutorial/java/index.html>

variables <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html>

data types <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

relational operators <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/op2.html>

if-then <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>

java math library <http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html>

enumeration types <http://docs.oracle.com/javase/tutorial/java/javaOO/enum.html>

Simple Video on BlueJ Debugger <http://www.youtube.com/watch?v=LUUPTbWV0g8>

**Some helpful tips:**

1. Compile often – do it.
2. ***Perform the tasks by hand to verify your work.***
3. Make sure to test ALL parts of your if statements. You will have to create your own tests for if statements etc …
4. There is a link to enumeration types shown above. Read this tutorial or look in your book on for information about enumeration types.
5. I repeat, you will need to understand how to use enumeration types for this lab. Study them, and look up sample code for how to use them. An enumeration type is basically a new data type that you are creating.

## Tasks: Follow the directions below to complete your lab assignment

Create a new project, and then add a class named “Lab10Tester”. You will need a single method in this class, your main method. A sample starter Lab10Tester.java has been created for you on Google Drive.

Complete the following tasks. The problems from the book are copied at the end of this document for your reference. I have made slight changes to the problems listed in the book.

***Task01***: Problem P5.2 from Book. Write a class called TaxReturn. You will need to have two instance variables to represent marital status and income. Marital status should be an Enumeration type, and income should be a double. You should create a constructor that accepts initial values for these two instance variables. You should create a single method in this class called getTax(). This method will have no parameters, and return the amount of the tax. (if marriage status is married, then the income represents the sum of both parties income. i.e. if harry makes 10, and sally makes 20, and they file as a married couple, then you would use “30” as the income value. TaxReturn.java has been started for you on Google Drive. Add tests in your main method that covers the code, and tests boundary conditions. You should print both expected and actual values.

***Task02***: Problem P5.9 from Book. Write a class called AutomaticTeller. You should have a single instance variable to store the PIN number for this ATM as an int. (do NOT assume you’re your PIN is 1234 as stated in the problem in the book). Create a constructor to accept an initial value for your instance variable. Create a method called validateUser() that accepts no parameters and has no return value. This method should prompt the user for a PIN, and then function as described in the book (you will do all user input of PIN attempts, and do all printing from this method for this task). For your screenshot of output you should make sure to test all cases. (i.e. PIN attempt correct on the first try, PIN attempt correct on 2nd try, etc …).

***Task03***: Problem P5.14 from the Book. Write a class called CarSimulator. Create a method in this class called checkDoors(). The method header is given for you below.

**public** **String** checkDoors(**int** leftDashSwitch, **int** rightDashSwitch,

**int** childLock, **int** masterUnlock, **int** leftInside,

**int** rightInside, **int** leftOutside, **int** rightOutside, GearShift gear)

Your method should return one of four possible String objects.

1. "Both Doors Closed"
2. "Left Door Open"
3. "Right Door Open"
4. "Both Doors Open"

There is a diagram that we will cover in class included on Google Drive that explains this method (task03.jpg).

Your main method should have several tests for this method (you do not need to test ALL combinations of inputs, as this would be over 256 combinations). Your main method should print both the expected value, and the String that is returned by the method checkDoors() which is your actual value.

***Task04***: Take a capture of your final output, which should show output for Task01, Task02, and Task03.

## To Turn In via Google Drive

You should turn in your java files and a document with your output.









