

Lab 6

Due Dates: In class portion 10/15 (Monday) 10/17 (Wednesday)
Out of class portion 10/24) all

1. **Correct Change** You did this problem in Visual Logic. Write a program to assist a cashier with determining correct change. In the java application we will use 2 classes, one a CorrectChange class and a second CorrectChangeTester.

In class

The CorrectChange class should do the following:

- a. Create instance variables, change, quarters, dimes, nickels, and pennies.
- b. Create constant variables to hold the value for quarters (.25), dimes, nickels, and pennies.
- c. Have a constructor that accepts as input (the explicit parameter) the value for change. The constructor should use the value from the explicit parameter as the value for the instance variable, change.
- d. There should be a method that determines the appropriate change in quarters, dimes, nickels and cents. (Hint: If you divide an integer by an integer you get an integer and use the java mod symbol, %, to get the remainder.)
- e. You should have a method that returns the number of quarters, another method that returns the number of dimes, a method that returns the number of nickels, and a method that returns the number of pennies.

The tester class should do the following:

- a. Use Scanner to ask the user for the change value. (Remember you must create a Scanner object and use the appropriate methods.)
 - b. Create an object of type CorrectChange.
 - c. Execute the method to determine the appropriate change.
 - d. Print the number of original amount, the number of quarters, dimes, nickels, and pennies.
2. In this problem you will develop an application for Employees. An employee has a name (a string) and a salary (a double). You will create an employee and then give them a raise. There will be two classes. An Employee class and an EmployeeTester class.

Out of Class

The Employee class should do the following:

- a. Create instance variables to hold the name (a string) and the salary (a double).
- b. Provide a constructor with two parameters to initialize both the name and the salary.

- c. Create a method, raiseSalary, to raise the salary by a given percentage which is received by the method through the explicit parameter.
- d. Create a method to return the name.
- e. Create a method to return the salary.

Create a tester program

- a. Create a Scanner object to use in reading input.
- b. Have the user input the employee name.
- c. Have the user input the employee salary.
- d. Print the employee name and salary.
- e. Create an employee object.
- f. Have the user input the percentage by which the employee will get the raise.
- g. Apply the raiseSalary method.
- h. Print the employee name and new salary

3. Write a class Bug that models(simulates – you do not draw the bug moving) a bug moving along a horizontal line. The bug moves either to right or left. Initially, the bug moved to the right, but it can turn to change its direction. In each move, its position changes by **one** unit in the current direction. You will have two classes, a Bug class and a tester class. **In class**

The Bug class should do the following:

- a. Create an instance variable, currentPosition, to represent the current position of the bug.
- b. Create an instance variable to represent the direction of movement (positive to the right and negative to the left). Initially it should be positive 1.
- c. Create a constructor that accepts as input (explicit parameter) the current location of the bug. The constructor should initialize the currentPosition.
- d. Create a method to turn the direction of the bug. (Hint: How can you change the direction from positive to negative mathematically?)
- e. Create a method to move the bug. Remember the bug only moves one position at a time.
- f. Create a method to return the current position of the bug.

Your BugTester class should do the following:

- a. Create a Scanner object to use in input.
- b. Ask the user to input the current position of the bug.
- c. Create a Bug object.
- d. Make the bug object move and turn several times.
- e. After each move print the actual and expect position after that move.

4. Your weight is actually the amount of gravitational attraction exerted on you by the Earth. Since the Moon's gravity is only one-sixth of the Earth's gravity, on the Moon you would weigh only one-sixth of the Earth's weight. Create an application that calculates your weight on each of the planets. **Out of Class**

Create a PlanetWeightCalculator class. This class will be a static class so you will not create an object and will not have a constructor. The class should do the following.

- a. Create a series of methods to calculate an individual's weight on various planets. Use the table below for the correct conversion. **Each** method should receive as input through the explicit parameter the user weight. Each method should be a static method. Each planet/moon calculation should be a separate method and should return the calculation.

Planet/Moon	Multiply the Earth Weight by
Moon	0.1666
Mercury	0.4
Venus	0.9
Jupiter	2.5
Saturn	1.1

The tester class should do the following.

- a. Create a Scanner object to use for input.
- b. Ask the user for their weight.
- c. Print the user weight.

- d. Call each of the planet methods and print the weight. .Make sure your output is such that the original weight as well the planet/moon weight is identified sufficiently to know what each belong to. Use the mathematical round method to round the outputted value.