**Name:**

**Advanced Programming in Java**

**Lab Exercise 12.11.2024**

1. A line in a plane can be specified in various ways:

* By giving a point (x, y) and a slope m
* By giving two points (x1, y1) and (x2, y2)
* As an equation in slope-intercept form y = mx + b
* as an equation x = a if the line is vertical

Implement a class Line with 4 constructors, corresponding to the four cases above. Implement methods.

boolean intersect(Line other)

boolean equals(Line other)

boolean isParallel(Line other)

Implement a LineTester class to test the Line class.

1. Implement a PermutationLock class. A permutation lock has a dial with 26 positions labeled A to Z. The dial needs to be set 3 times in the correct order. If it is set to the correct permutation, the lock is opened. When the lock is closed again, the permutation can be entered again. Write a LockTester class to test the PermutationLock class.

Note: A permutation is a sequence where order matters. A combination is a sequence where order does not matter. The traditional combination lock is really a permutation lock.

1. Create a Delivery class for a delivery service. The class contains fields that holds the following:

* A delivery number that contains 8 digits. The first four digits represent the year, and the last four digits represent the delivery number. For example, the 76th delivery in 2011 has a delivery number of 20110076.
* A code representing the delivery area. A local delivery is code 1 and a long-distance delivery is code 2.
* A weight, in pounds, of the item to be delivered.
* The fee for the delivery as follows:

|  |  |  |
| --- | --- | --- |
| Distance | Weight | Fee($) |
| 1 | Under 5 pounds | 12.00 |
| 1 | 5 to 20 pounds | 16.50 |
| 1 | Over 20 pounds | 22.00 |
| 2 | Under 5 pounds | 35.00 |
| 2 | 5 pounds or more | 47.95 |

Create a constructor for the Delivery class that accepts arguments for the year, delivery number within the year, delivery distance code, and weight of the package. The constructor determines the eight-digit delivery number and delivery fee. Also include a method that displays every Delivery object field.

Next, create an application that prompts the user for data for a delivery. Keep prompting the user for each of the following values until they are valid:

* A four-digit year between 2001 and 2025 inclusive
* A delivery number between 1 and 9999 inclusive
* A package weight between 0.1 pounds and 100 pounds inclusive
* A delivery distance code that is either 1 or 2

When all of the data entries are valid, construct a Delivery object and then display it’s values.

1. **Armstrong numbers**

An Armstrong number is a whole number that’s equal to the sum of its digits raised to the power of the total number of digits. For example, 153 is an Armstrong number because there are three digits, and 153 = 13 + 53 + 33. The four-digit number 8208 is also an Armstrong number, as 8208 = 84 + 24 + 04 + 84.

Create an Armstrong number checker that returns a Boolean TRUE if the input number is an Armstrong number. Hint: to extract each digit from a given number, try using the remainder/modulo operator.

If you’re looking for something a little more challenging, create an Armstrong number calculator that returns all Armstrong numbers between 0 and the input number.