CITP 190 – Introduction to Java
Project 11

This project requires you to create one project. The project will work with exceptions. Please be sure to download and unzip the project start files.

To receive full credit for this project you must submit the following:

- A detailed design diagram for each source code file.
- The source code files (the .java files) following the coding standards.
- The bytecode for each source code file (the.class file)
- Proof of the correctness of your output using the test data provided. Please note that you must provide proof for all test data. You may provide additional test data.
- A capture of the output of the program. The output must show all data from your proof.

Submit all files as one (1) ZIP file to the Project 11 Drop Box in the course site.

Grading:

Program design	5 points
Design diagrams	5 points
Following course standards	5 points
Proof	3 points
Screen captures	2 points

Important notes:

1. Incorrect calculations will result in a 0 grade for this project.
2. Output that is not presented as shown (including spaces and spelling) will result in a 0 grade for this project.
3. Code that does not follow the standards will result in a 0 grade for this project.
4. Uploading more than one zip file will result in a 0 grade for this project.
5. Not using all the test data provided will result in a 0 grade for this project.

Project 11: Display customer information

Console

```
Welcome to the Customer application

Enter a customer number: 1003

Ronda Chavan
518 Commanche Dr.
Greensboro, NC 27410

Display another customer? (y/n): y

Enter a customer number: 2439

The customer 2439 does not exist.

Display another customer? (y/n): n
```

Operation

- This application displays customer information for customers selected by the user. The application prompts the user to enter a customer number. If a customer with that number exists, the application displays the customer's name and address. If no customer with that number exists, the application displays the message "The customer *number* does not exist." Either way, the application then asks if the user wants to display another customer.

Specifications

- Validate that the user has entered an integer between 1 and 5000 inclusive for the customer number. You may include a Validator class with the appropriate method.
- Use the Customer class that has been provided.
- To get the information for a customer, use the CustomerIO class that's provided. This class contains a method named `getCustomer` that accepts a customer number (an int value) and returns a Customer object.
- Modify the `getCustomer` method so that if it's called with an invalid customer number, it throws an exception of type NoSuchCustomerException. Do not make any other changes to the method.
- Create a NoSuchCustomerException class. This class should have a constructor that accepts an `int` parameter that provides the customer number that doesn't exist. This constructor should construct the message "The customer *number* does not exist." where *number* is the customer number. It should pass the message to the constructor of the Exception class. For example, if the user enters 500 as the customer number, the message would be:
  `The customer 500 does not exist.`
- The NoSuchCustomerException should store the customer number as a private instance variable and make it available through a method named `getCustomerNumber`. If the user enters an invalid customer number, the main application class should use the `getCustomerNumber` method to retrieve the customer number from the exception object when it displays the error message to the user.