

CITP 190 – Introduction to Java

Project 7

This project requires you to create one project. The project will work with customer and employee data. 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 method in each class.
- A class diagram for each class.
- The source code for each class (the .java file) following the coding standards.
- The bytecode for each class (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 7 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 7: Work with customer and employee data

Console

```
Welcome to the Person Tester application

Create customer or employee? (c/e): c

Enter first name: Frank
Enter last name: Jones
Enter email address: frank44@hotmail.com
Customer number: M10293

You entered:
Name: Frank Jones
Email: frank44@hotmail.com
Customer number: M10293

Continue? (y/n): y

Create customer or employee? (c/e): e

Enter first name: Anne
Enter last name: Prince
Enter email address: anne@murach.com
Social security number: 111-11-1111

You entered:
Name: Anne Prince
Email: anne@murach.com
Social security number: 111-11-1111

Continue? (y/n): n
```

Operation

- The application prompts the user to enter a customer or an employee.
- If the user selects customer, the application asks for name, email, and customer number.
- If the user selects employee, the application asks for name, email, and social security number.
- When the user finishes entering data for a customer or employee, the application displays the data that the user entered.

Specifications

- Create an abstract Person class that stores first name, last name, and email address. This class should provide a no-argument constructor, get and set methods for each piece of data, and it should override the toString method so it returns the first name, last name, and email fields in this format:

Name: Frank Jones
Email: frank44@hotmail.com

In addition, it should contain an abstract method named getDisplayText that returns a string.

- Create a class named Customer that inherits the Person class. This class should store a customer number, it should provide get and set methods for the customer number, it should provide a no-argument constructor, and it should provide an implementation of the getDisplayText method. The getDisplayText method should return a string that consists of the string returned by the toString method of the Person class appended with the Customer number like this:

Name: Frank Jones
Email: frank44@hotmail.com
Customer number: M10293

- Create a class named Employee that inherits the Person class. This class should store a social security number, it should provide get and set methods for the social security number, it should provide a no-argument constructor, and it should provide an implementation of the getDisplayText method. The getDisplayText method should return a string that consists of the string returned by the toString method of the Person class appended with the Employees social security number like this:

Name: Anne Prince
Email: anne@murach.com
Social security number: 111-11-1111

- Create a class named PersonApp that prompts the user as shown in the console output (this is the class that will contain your main() method). This class should create the necessary Customer and Employee objects from the data entered by the user, and it should use these objects to display the data to the user. To print the data for an object to the console, this application should use a static method named print that accepts a Person object.
- Use the Validator class or a variation of it to validate the user's entries. Do not attempt to validate the format of the email address or phone number. We have not yet covered that material.

Test Data

Menu Option	First Name	Last Name	e-mail	Customer Number/ Social Security Number
c	Terry	Smith	smitht@abc.com	A458
w	Chris	Jones	chrisj@yahoo	555-12-5896
e	Sean	Walters	Crazy.rider@hotmail.com	555-88-9632
c	Tiki	Baba	Babat@xyzcorp	X752

Note: The test data above contains an error. In your proof, note the error and the correction you will use when running the program.