CECS 220
Summer 2019
Assignment 5
July 16, 2019

Due Date: Sunday, July 21, 2019, 23:55
Assignment 5 Points: 20
Bonus Points: 10

1. **(20 points)** Solve the following problem (Ch. 11).
   Write a program to enter employee data including his or her name, Social Security number, and salary, into an array. The maximum number of employees is 100, but your program should also work for any number of employees less than 100. Your program should use two exception classes, one for the case when the Social Security number length is not exactly nine characters (without dashes or spaces), and another when the Social Security number includes any character that is not a digit, as defined below.

   (a) Create a class called *Employee* which has three instance variables, *Name*, *Salary*, and *SSNumber*. The *Employee* class has one constructor that accepts values for initializing the *Name*, *Salary*, and *SSNumber*. Include accessor methods, mutator methods, equals method and toString method. The constructor method, or the mutator method for the *SSNumber*, should throw exceptions when a Social Security number does not satisfy the required format. The equals method returns true if the two employee objects have either the same name or the same Social Security number, and false otherwise.

   (b) Define two exception classes. One is called *SSNLengthException*, for the case when the Social Security number entered without dashes or spaces is not exactly nine characters. The other is called *SSNCharacterException*, for the case when any character in the Social Security number is not a digit. The two exception classes should display appropriate messages, when they are thrown, telling the user what has been entered and why they were not appropriate.

   Write a client class, called *EmplyeeBuilder*, that builds a list of *Employee* objects, using an array of maximum size of 100. The client allows a user to enter values for an employee. The client class catches and handles the exceptions if they are thrown due to creating unacceptable *Employee* object that does not satisfy the Social Security number format. Besides, the program should not allow two employees of the same name or Social Number to be added to the built list of Employee objects. No employee record should be added to the list if it does not satisfy the conditions. After all data has been entered, your program should display the records for all employees, with an annotation stating whether the employee's salary is above or below average.

2. **(10 points, optional bonus problem)** Write a JavaFX application that calculates the registration fees for a conference. The general conference fee is $800 per person, and student registration is $450 per person. There is also an optional opening night dinner with a keynote speech for $30 per person. In addition, the optional preconference workshops listed below are available.

Workshop	Fee
Introduction to E-commerce	$350
The Future of Web	$300
Advanced Java Programming	$400
Network Security	$450

The application should allow the user to select the registration type using two radio buttons, the optional opening night dinner using a check box, and one preconference workshop as desired using either a choice box, a spinner, or a list view. The total cost should be displayed on the screen window.

Submission Requirements:

Your presentation in your report reflects great deal about you, your understanding of the assignment and on how much this course means to you. I try very hard to look at the substance of the report but I will be lying if I said that presentation does not influence my judgment. So, I expect your reports to be well organized and conform to the following rules:

All reports must be submitted in PDF format. Each assignment should contain the following:

1. Title page with your name, assignment number and the day you are actually submitting this report (Not the assignment due date).
2. A brief description of your solution of each problem of the assignment separately, you can also explain your solution using a pseudocode. *Number your descriptions according to the problem numbers.*
3. A comprehensive set of snapshots showing the inputs submitted, outputs obtained in the case of a successful output or a failure, including required output formatting, prompts, and messages.
4. Java source files that contain your solutions. It must be a "*.java" file. Source programs should contain meaningful comments and variable names.
5. Please zip both the PDF document with the source code and submit one zipped file. Please name your zipped file as "HWx_firstname_lastname.zzz". Where, "firstname" and "lastname" refer to your first and last names, "x" refers to the homework number (e.g., 1, 2, etc), "zzz" refers to the file name extension for the software used for archiving.
6. **Submissions after the due date are accepted with a penalty of 25% per day (weekend days are counted as one-day delay).**

Grading Table Summary

Problem	Item		Points
Problem 1 (20 points)	Report (Description of solution, and pseudocode)		2.0
	Output snapshots (input/output, formatting)		1.0
	Java Source Code Implementation	Exception classes	1.0
		Employee Class	9.0
		EmployeeBuilder Class	7.0
Bonus Problem 2 (10 points)	Report (Description of solution)		1
	Output snapshots (input/output, formatting)		1
	Java Source Code Implementation	Correct implementation satisfying requirements	8
	Total		30