

Lab 11: Adding Persons to Collections

1. Make the following modifications to your person application:
 - a. Create an enum for person type in class Person to hold values for all valid subtypes
 - b. Modify the toString method of Instructor to display "Professor" in front of the first name.
 - c. In the driving program, prompt for whether the person is a Student or Instructor. To control the input, use a Swing list control. Pass the control the possible enum values. Look to <http://jackmyers.info/java/src/Lesson-07/ComboBoxShowInputDialogExample.java> for a model on how to use Swing in this manner.
 - d. Add the persons you are creating to the person collections below. Be sure to include a mix of Students and Instructors.
 - i. ArrayList, LinkedList, HashSet
 - e. Create a method that will display the elements of your sets. The display should look like this: (Again, for ease, you can hardcode the Student status and major as well as the Instructor department.)

Full Display

Marshall J Peterson
Student
DOB: 19-Sep-1984 (Age: 28)
eMail: petersonm@students.acme.edu
SSN: 793-50-9204
Status: NONMATRICULATING
Major: ART

Professor Anthony M Sommers
Instructor
DOB: 02-Jul-1961 (Age: 51)
eMail: sommersa@acme.edu
SSN: 747-16-3296
Dept: ENGINEERING

Darryl Jackson
Student
DOB: 01-Apr-1991 (Age: 21)
eMail: jacksond@students.acme.edu
SSN: 202-25-8714
Status: UNDERGRADUATE
Major: ART

Name Display

Indira J Phillips
Vera M Arnold
Vera J Peterson
Randall O Arnold
Eileen E Ridgeway
Oliver Henderson
Geraldine A Moore
Marshall J Williams
Penny A Sommers
Geraldine J Williams
Larissa Williams
Indira C Jackson
Oliver Williams
Tara E White
Franklin T Goldschmidt
Professor Bill E Williams
Professor Indira C Goldschmidt
Professor Sylvia O Ridgeway
Professor Jane J Shapiro
Professor Larissa A Wu
Professor Marshall M White
Professor Randall J Singh

2. Object Equality

For this section, you might not want the annoyance of constant prompting to enter Person data. Assuming you completed part 1, you may clone your driver program and hard-code the additions of Students and Instructors into your various Collections. I uploaded a random Person generator that I used for this. Techniques like a random data generator are very useful in practical programming during testing. See http://jackmyers.info/java/src/Lesson-05/5-12/Sample_random_Person_data_generation_method.txt for the example I used. If you use it, you may need to modify it based on your implementation.

- a. Build the empty constructor for Person if you have not already done so.
- b. Add a constructor to Person so that you can create a person from another Person. This is called *prototyping*, and is a feature of some modern object-oriented languages like Self, Lua or Io. In this way you can use an existing person to create a new person and then just modify the properties that differ.
- c. Remembering that constructors are the only non-inherited methods, create similar constructors in Student and Instructor.
- d. Overriding the equals() method.

Now create a method in Person that overrides the equals() method. Recall that to override you have to have the same method signature in terms of data type and arity. What do you suppose the method signature of equals() is?

Person's equals() method should use the unique value Social Security Number to determine equality. However, in our implementation, we will allow an Instructor to also be a Student. Therefore, let's encode the following algorithm:

- Two Person objects will be equal if:
 - i. They have the same SSN.
 - ii. They are of the same object type.
(e.g., 198-22-9999 the Student is NOT equal to 198-22-9999 the Instructor.)

(Hint: You might need to use a method to determine the class of an object.)

- e. Testing equality

With your new constructor, you can retrieve one of your Persons from a Collection and use it as a template to instantiate another Person object. Test ==, equals(), and the collection method contains(). How do these operations differ?

3. One Last Collection

Implement TreeSet as your fourth collection. What extra requirement does this impose on your system? (Hint: You will need to implement an interface.)