**IT 2045C Computer Programming I I  
Prof. Tom Wulf   
Lab 01 Get Classy  
Spring Semester 2018**

Learning Goals:

* Get practice creating classes with Netbeans
* Instantiate object data from a text file
* Persist object data to a text file
* Store and manipulate class objects with an ArrayList

**Functional Requirements:**

* Create each of the classes below as a separate Netbeans project.
* Create a constructor that takes all the fields
* Create additional overloaded constructors where it makes sense to do so
* Create getters for all fields.
* Create setters for fields where it makes sense to do so
* Create additional specified class methods
* In each case create a java main unit test class for your class
  + Use the expected pattern where the test will show the expected output
  + Test thoroughly!
* Use the project and file names specified

**Mini-lecture:**Last week, we reviewed how to save and read data records to and from a file. This week we are looking at how to create objects in java, so we will create java classes that correspond to our data records, **Person** and **Product**. We will modify the programs from the previous lab to use our new object classes.

**1: Person**

Project: Person  
Files: Person.java   
 PersonReader.java // Reads Person records from a file into an ArrayList  
 PersonGenerator.java // Creates Person objects into an ArrayList and writes it to a text file  
 SafeInput.java

Fields:

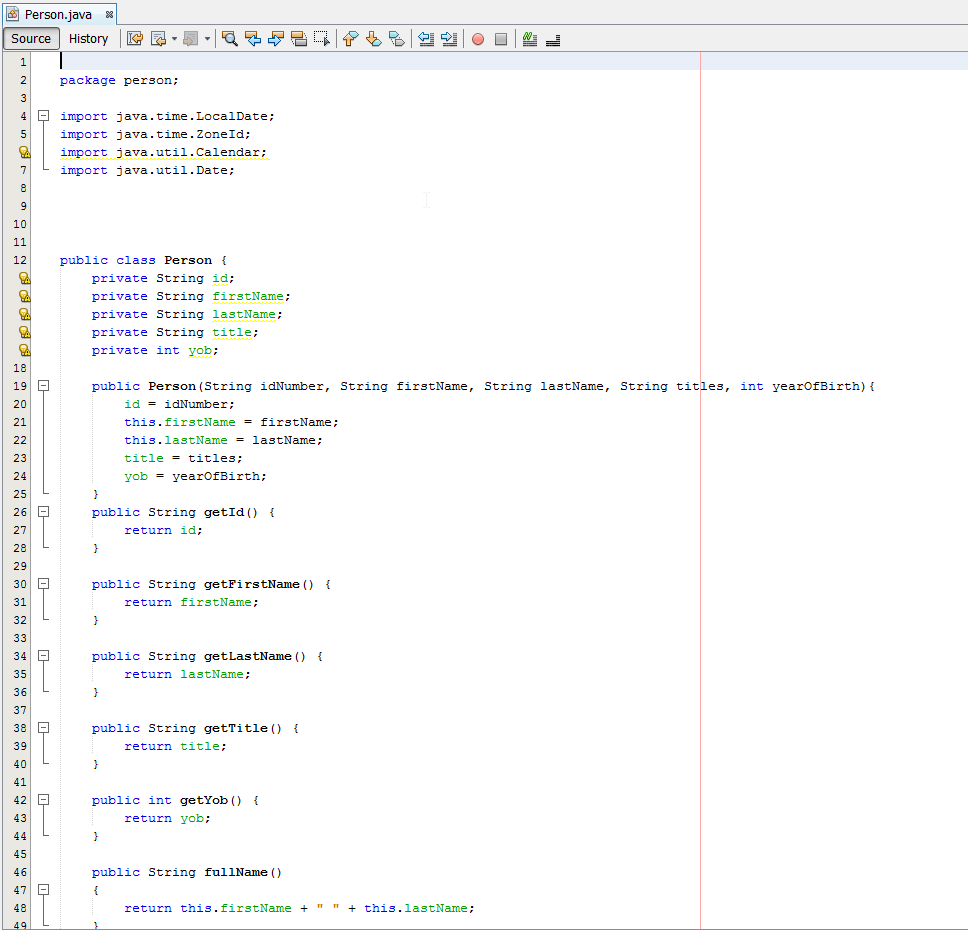
String firstName  
String lastName  
String ID // should never change sequence of digits  
String title // a prefix: Mr. Mrs. Ms, Prof. Dr. Hon. Etc.  
int YOB // Year of birth // Range should be 1940 - 2000

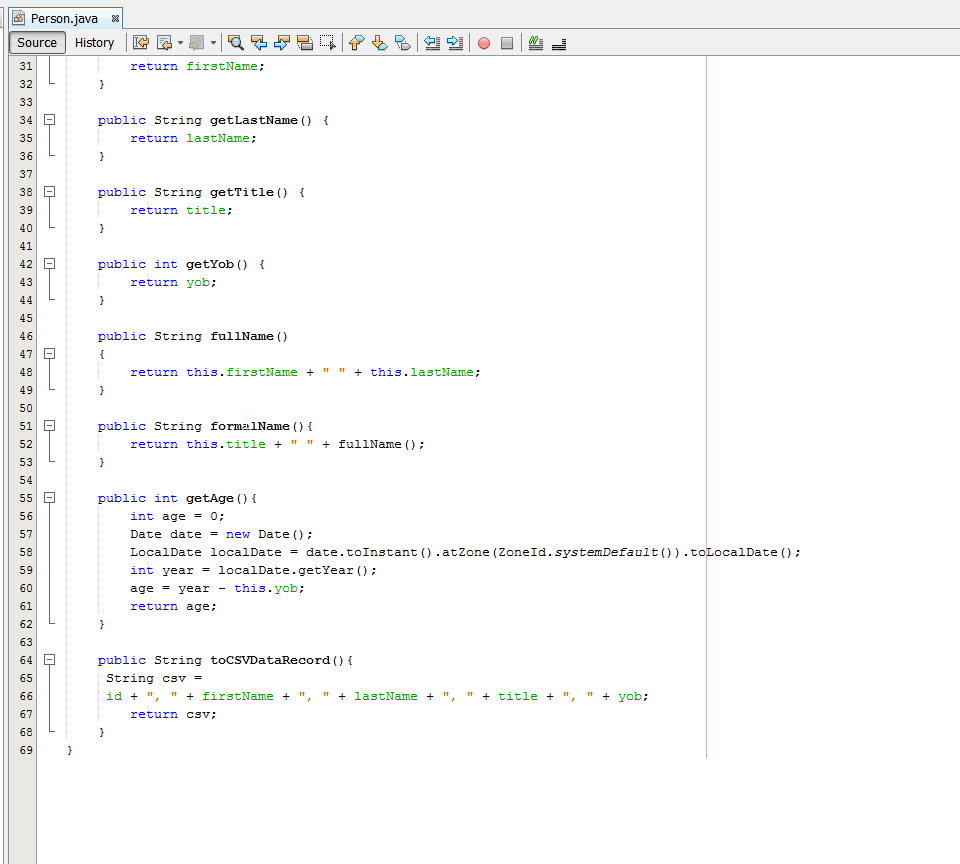
Additional methods:

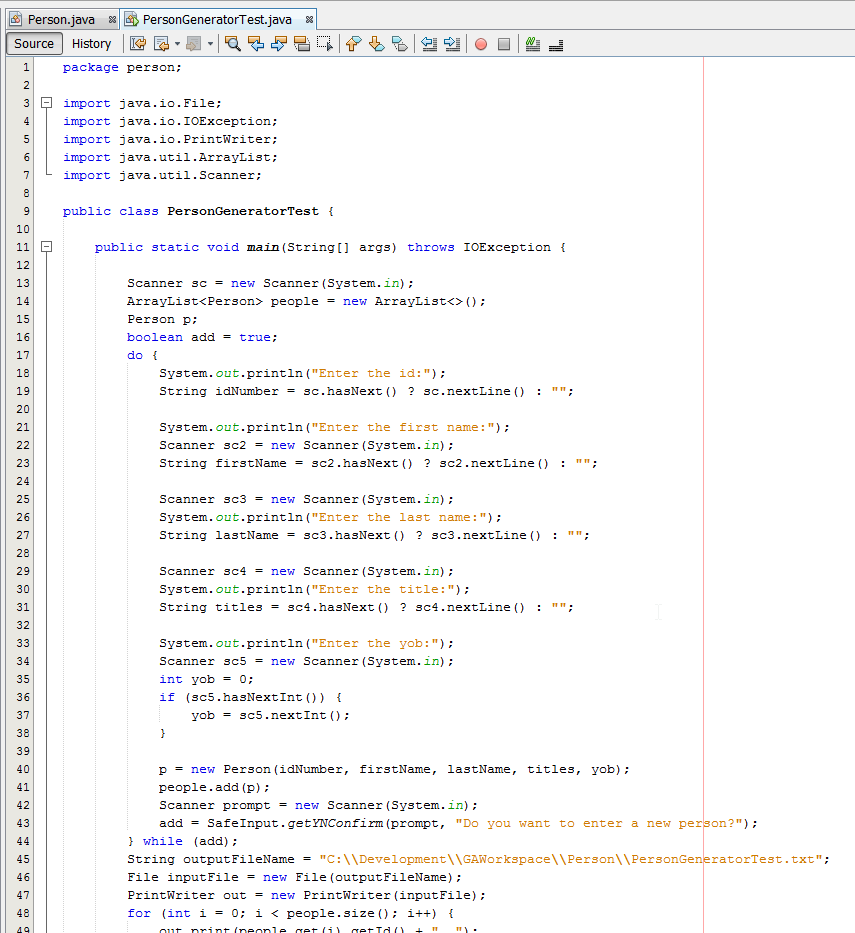
public String fullName() // returns firstName, space, lastName  
public String formalName() // returns title, space, fullName  
public String getAge(int year) // uses YOB to calculate age for specified year use the Calendar object to do this. Requires a bit of a web search.

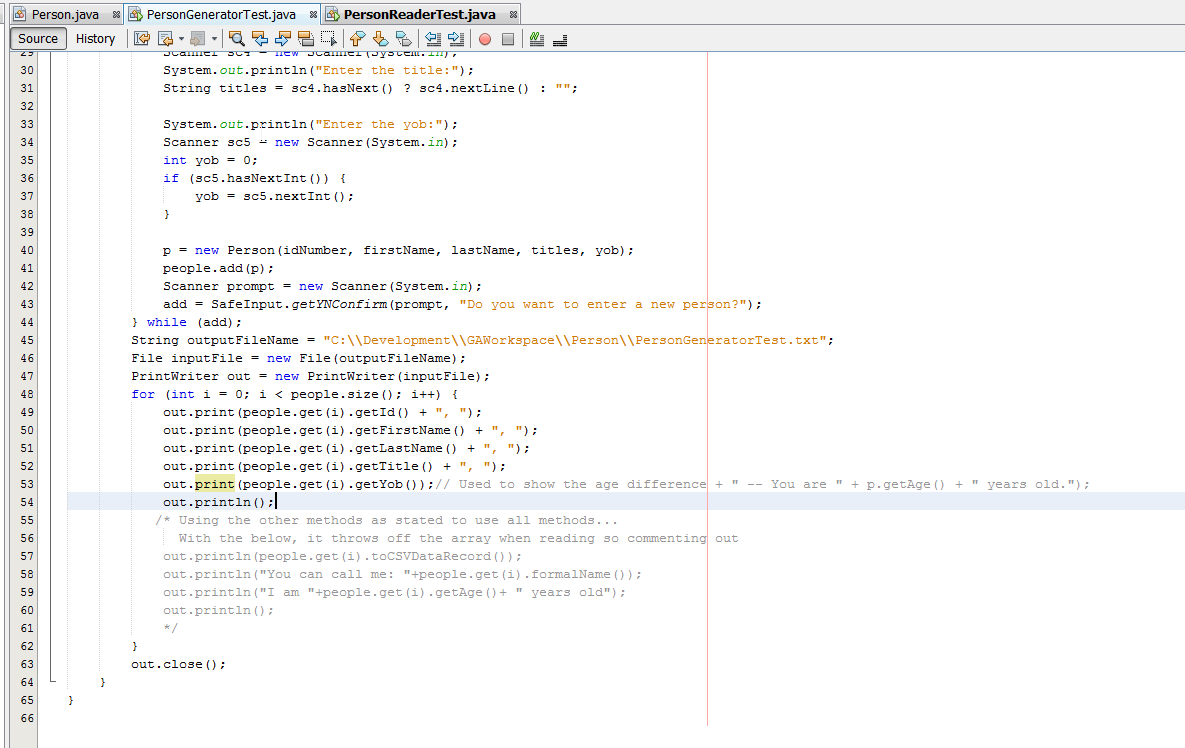
Public String toCSVDataRecord() // returns a comma separated value (csv) String suitable to writing to a java text file. Be sure to use this function when you save data to the file.

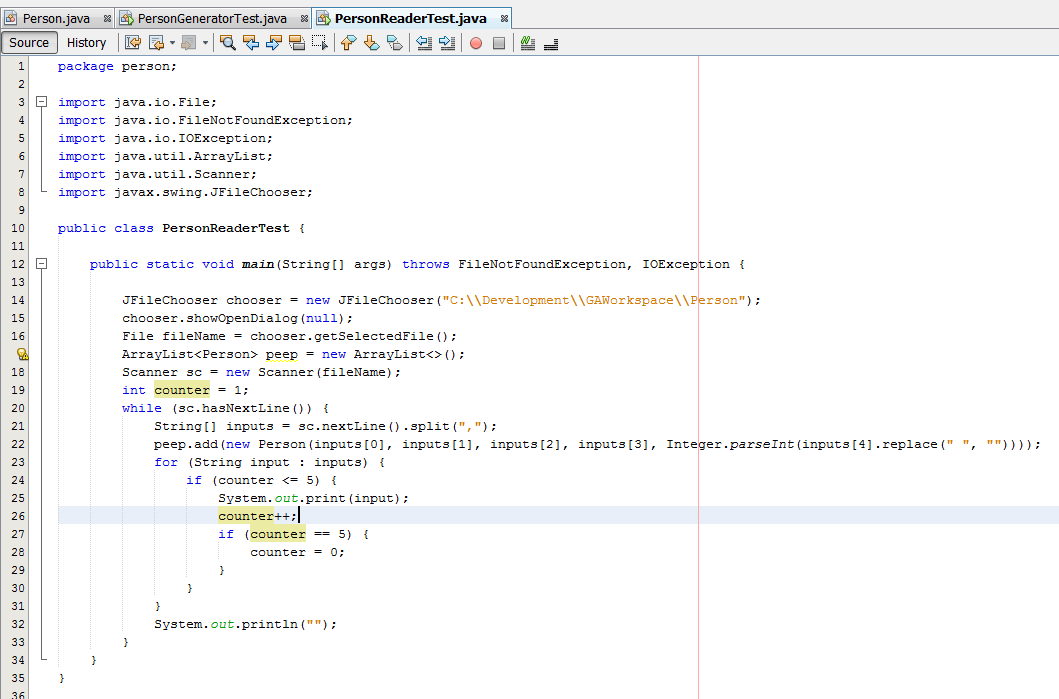
When you read the data in from the textfile, instantiate a Person object from each record and store the Person objects within an ArrayList as type <Person>. Similarly, when in PersonGenerator as the user enters the data fields for each person, use it to create a Person Object and again store it in an ArryList and then write it to the text file when they are finished.

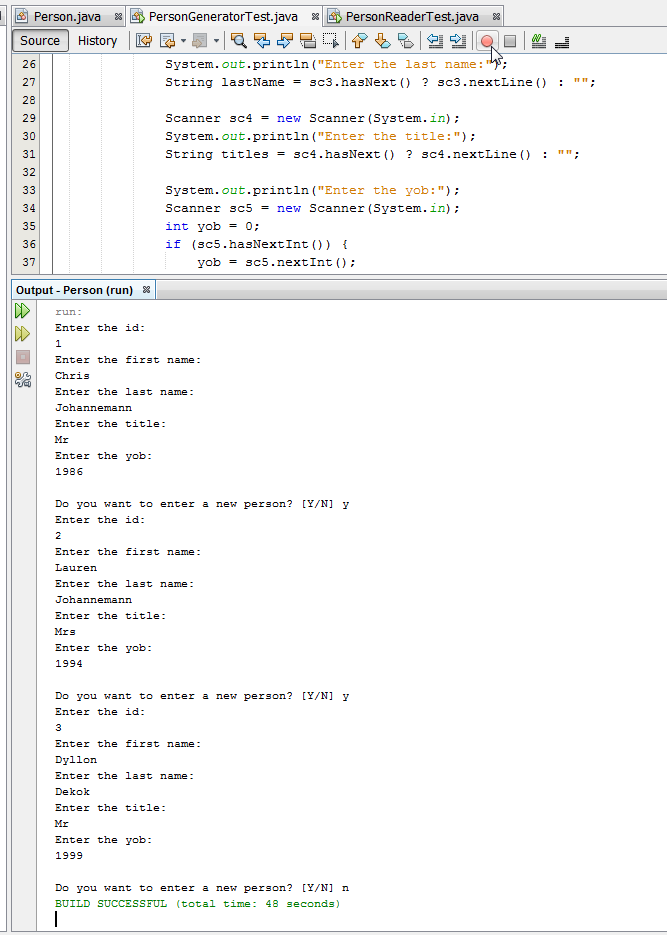
Make sure that your program uses all the methods here.  


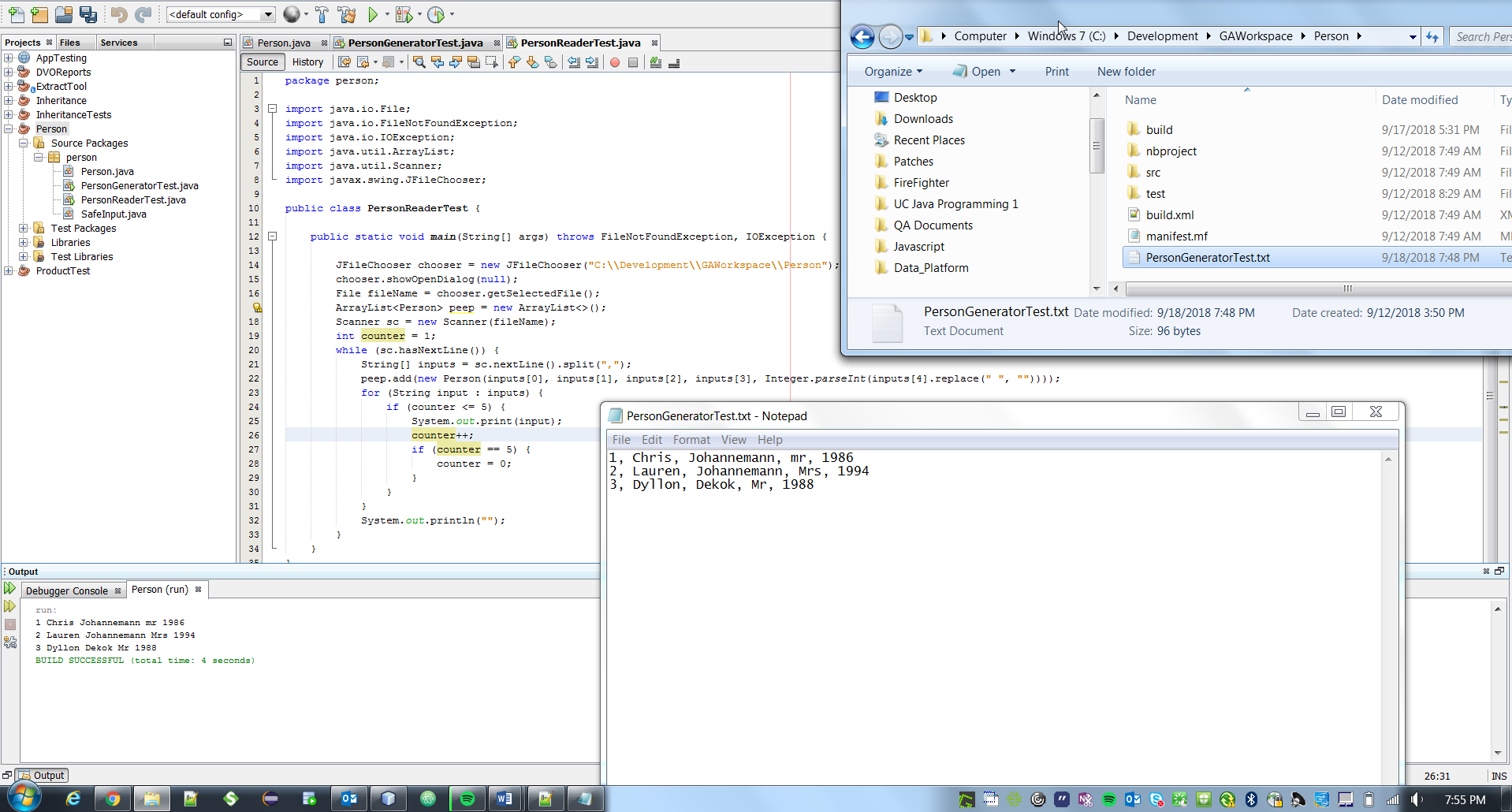












**2: Product**

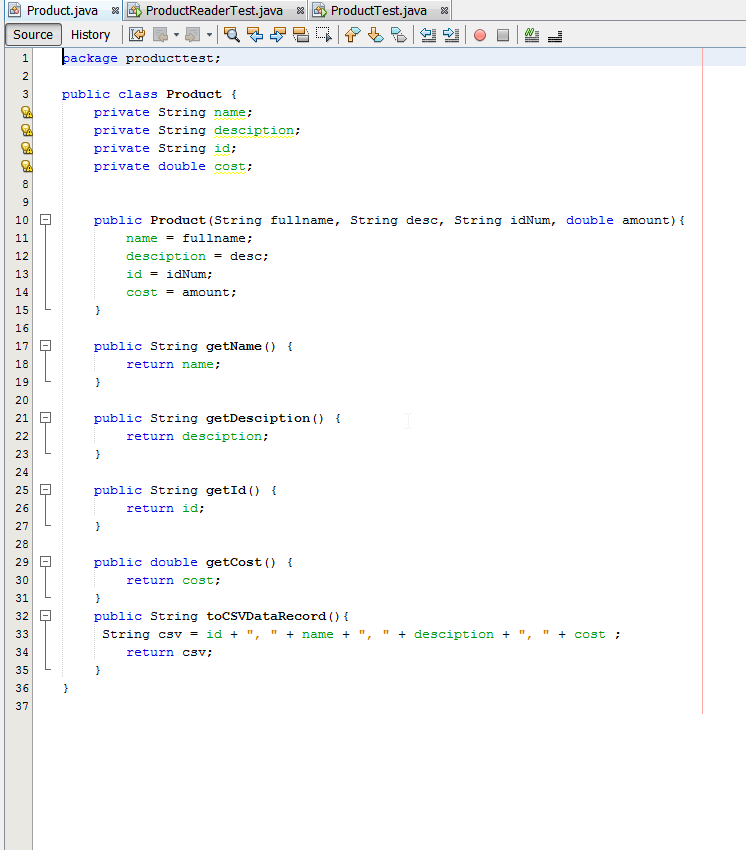
Project: Product  
Files: Product.java ProductTest.java

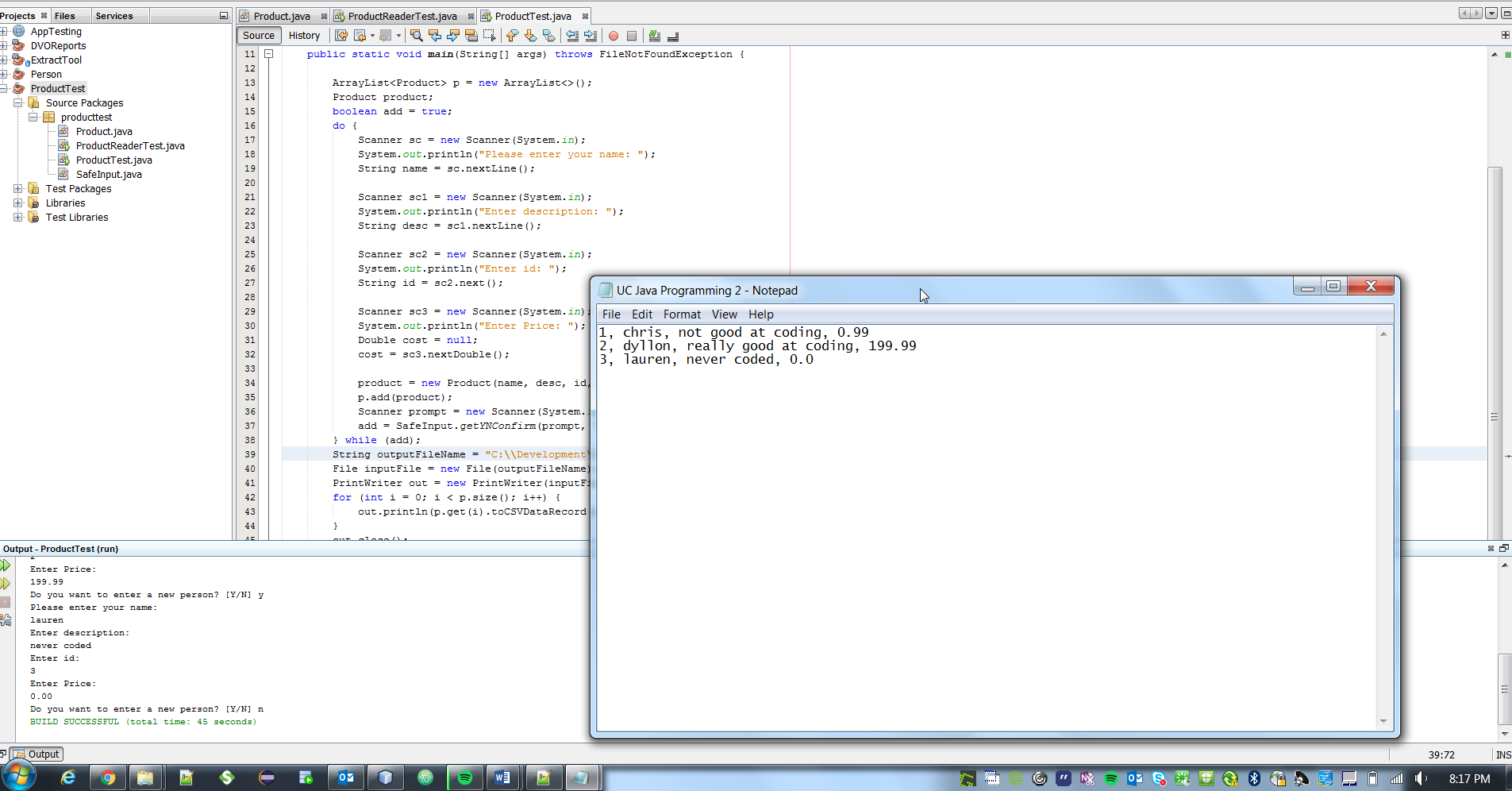
Fields:

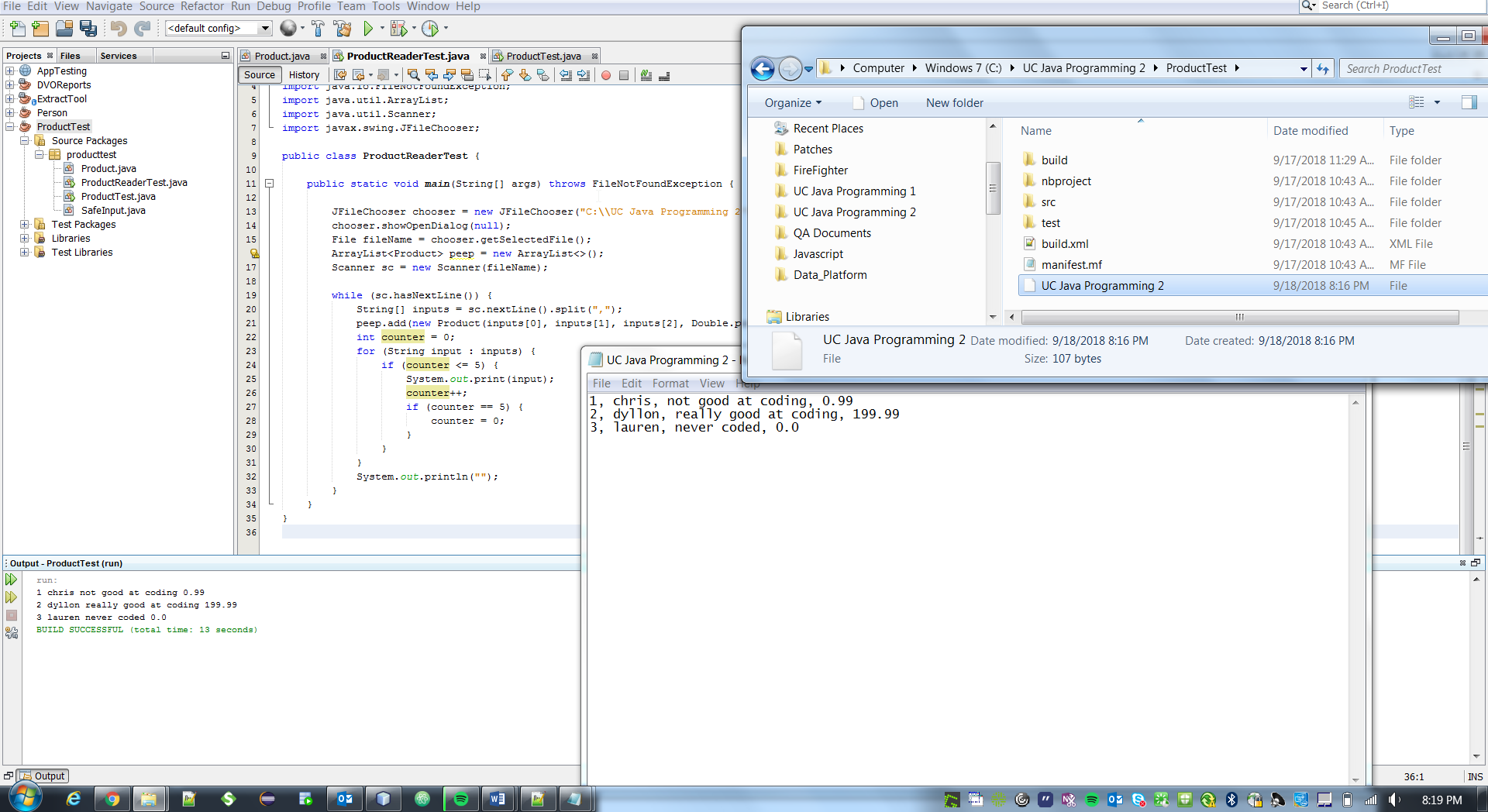
String name  
String description  
String ID // should never change   
double cost

Do the same thing with Product as you did with Person.Screen shots: record screen shots of your output for each of your classes at the end of this file.

* Creating/Saving Person and Product data
* Displaying Person/Product data after reading the file.







**Submission:**

Create a single zip archive called **Lastname\_Firstname\_Lab01.zip** that includes each of the complete Netbeans project folders for your classes. Add this file **Lastname\_Firstname\_Lab01.docx** with you screen shots to the archive. Submit your zip archive and separately submit **this docx file** with the screen shots. Then I can read it directly in Bb.  
  
Can’t follow directions? I can’t be bothered to deal with you so I won’t grade your submission.  
I won’t volley back and forth with you to train you to get it right. You get one free ride on this.