**CS1401 – Programming Assignment 9**

**Due: Monday November 24, 2015 by 8:00 a.m. for those of you taking the MW labs**

**Tuesday November 25, 2015 by 8:00 a.m. for those of you taking the TR labs**

**To be submitted via Piazza in the folder “lab9”**

**Welcome to Lab 9!**

|  |
| --- |
| This week and next, you will get to practice on user-defined types. You will have to implement a new type (class) and use it. Using your new type will be yet another opportunity to practice your skills about methods.  As usual, we hope you have fun!  Now, let’s get started! |

** You should expect to work about 6 to 8 extra hours outside the lab session to complete this assignment.**

**It means that you need to make sure that Java works on your own computer, or that you go to open labs to work some more.**

**Extra time on labs includes completing the activities and taking the time to make sure that your submission is picture perfect!**

**When you can’t find it, you build it! (User-defined types)**

As a video-game storeowner, you want to inventory your games, to be able to order your games by bestseller, to compute your profit, etc. However, it is not always easy to do so without a new type of objects to manipulate.

In this lab, you are expected to implement the following:

1. **Design a class VideoGame** (in a file VideoGame.java) that has the following list of attributes (all of them private):
   1. A name: of type string
   2. A price: of type double
   3. A number of items sold: of type int
   4. A type of game: of type string

This VideoGame class provides you with a blueprint to create VideoGame objects.

1. To complete the class VideoGame, you will also be required to implement the following class methods\* (these are non-static methods):
   1. Constructors (essential to create instances of your blueprint VideoGame)

You will implement two constructors: the default constructor as well as a constructor that takes 4 parameters (a string – the name of the game, a double – the price of the game, an int – the number of items sold, and a string – the type of game)

* 1. Accessors: one method per attribute so that the value of each attribute can be accessed.
  2. Setters / Mutators: one method per attribute so that the value of each attribute can be set / modified.
  3. Additional method: Method Print:

For a given VideoGame where:

* + - * name is MyVideoGame;
      * price is $50.95;
      * number of items sold is: 10; and
      * type of game is: First-Person Shooter

this method will result in printing the following:

**Video Game MyVideoGame is a First-Person Shooter game. Its price is $50.85, and so far, 10 items have been sold**.

*\*some of the above methods are already provided to you in VideoGame.java to “show you the way”.*

1. You will then implement **another class, MyStoreManager**, in which you will implement the following methods:
   1. ReadFromFile: this method takes the name of a file as parameter: the given file contains information about video games. It reads the information and fills an array of videogames with this information. It returns the array of video games.
   2. SortBySold: this method takes an array of video games as an input and sorts it by the number of items sold of each video game (in descending order of number of items sold). Use the same selection sort algorithm as you implemented in lab8 – just modify it slightly.
   3. SortByPrice: this method takes an array of video games as an input and sorts it by the price of each video game (in descending order of the prices). Use the same selection sort algorithm as you implemented in lab8 – just modify it slightly.
   4. SortByType: this method takes an array of video games as an input and sorts it by the type of each video game (in ascending alphabetical order). Use the same selection sort algorithm as you implemented in lab8 – just modify it slightly.
   5. TotalSales: this method takes an array of videogames and returns the total amount of money gotten from sales of the video games.
   6. Lottery: this method takes an array of video games and returns a randomly picked video game.
   7. The main method in which you test all the above methods.

In addition to testing each of your methods, in the main methods, you should implement the following scenario:

* + - * You ask the user for a file name
      * You read the file and retrieve information about videogames: build an array of VideoGame objects
      * You print out the result of the lottery and print out to the user its prize video game
      * You sort this array by number of items sold and from the resulting (modified / sorted) array, you print out the best seller video game
      * You sort this array by price and from the resulting (modified / sorted) array, you print out the cheapest video game
      * You sort this array by type of games and from the resulting (modified / sorted) array, you print out the number of games of each type
      * Based on the array of video games, you print out the total amount of sales

**What you have to turn in:**

* In a docx file, write the pseudocode of each of the above methods as well as a clear description of the new class VideoGame: remember that a good pseudocode can be handed out to a programmer without further explanation and the programmer should be able to translate it to code. *This is the standard that you will be held to.*
* In the java files that were provided to you, complete the description of the methods as described above and according to your pseudocode + include in the main method the lines of code that are necessary to meet the above specification.

**Important Note:** If your code is not properly indented, it will not be graded (you will receive 0 for the coding part of your grade).

**Important notes:**

* Indent your code properly following guidelines available at: <http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-136091.html>. Badly-indented code will be graded 0.
* Spend time working on your pseudocode as the amount of points you get for the pseudocode is bigger than the amount of points you get for your code (usually, close to a 60/40 ratio).

**That’s all! Looking forward to seeing you in lab!**