CSCI 220 -- PA 2

OOP – Classes, Inheritance, and Polymorphism

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab. You cannot work with a lab partner for this assignment.

You must use an appropriate provided template from Canvas or my website (zeus.mtsac.edu/~tvo) and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

Review important object-oriented programming concepts such as classes, inheritance, and polymorphism.

Exercise 1 – Perform C-2.5 from your C++ textbook (use Progression class from book as a base class).

Exercise 2 – Game Information.

Provide two classes, **GameEntry** and **GameScore**, to maintain a list of top scores according to the requirements below.

Each game entry (GameEntry) consists of a player name, a score (0 to 1000), and a date (mm/dd/yyyy). Include appropriate constructor(s), setters, and getters for class GameEntry. Do validate scores (set to 0 for an invalid score), but you can assume that dates are valid.

The game scores (GameScore) hold the name of the game and a list of up to 10 scores by using a list of game entries (a current list might have fewer than 10 entries, but no more than 10 entries). A new game entry can be added if applicable. A complete list of scores can be printed upon request (always from highest to lowest). In addition to appropriate constructor(s), setters, and getters, this class must support the following public operations: ***add(entry)*** and ***print()***.

Make sure to provide the two classes with basic operations described above and perform data validation as needed. Feel free to add additional classes and private operations as applicable. Write a test driver (main) to test your classes (separate from your implementation file). You can use a menu or code various operations in your test driver. Be sure to test that all operations to make sure they work properly.

Here is one example and you must try it as one test case:

```
GameScore g1("Classic Pac-Man");       // one game in C++
// GameScore g1 = new GameScore("Classic Pac-Man"); // Java
e1 = new GameEntry("Jill", 980, "08/05/2021"); // one entry
g1.add(e1);   // assume e1 (GameEntry): Jill 980 08/05/2021
g1.add(e2);   // assume e2 (GameEntry): Jack 600 08/18/2021
g1.add(e3);   // assume e3 (GameEntry): Rob  875 07/30/2021
g1.add(e4);   // assume e4 (GameEntry): Rob  900 08/02/2021
g1.print();   // print the following


Name: Classic Pac-Man
1  Jill     980   08/05/2021
2  Rob      900   07/30/2021
2  Rob      875   07/30/2021
3  Jack     600   08/18/2021
```

Question 1: What are some good reasons for taking advantage of inheritance?

Question: Explain the differences between static and dynamic binding.

Extra Credit (2 points)**:** Load scores from a text file. Here is the format for a sample data file (pacman.txt):

```
Classic Pac-Man
Jill  980   08/05/2021
Jack  600   08/18/2021
Rob   875   07/30/2021
Rob   900   08/02/2021
```

Driver:

```
// input game name from file
GameScore g1(gameName);    // one game in C++
// GameScore g1 = new GameScore(gameName); // Java

// input one entry from file and add until end of file
e1 = new GameEntry(player, score, date); // one entry
g1.add(e1);   // assume e1 (GameEntry): Jill 980 08/05/2021

g1.print();   // print the following


Name: Classic Pac-Man
1  Jill     980   08/05/2021
2  Rob      900   07/30/2021
2  Rob      875   07/30/2021
3  Jack     600   08/18/2021
```

Fill out and turn in the PA submission file for this assignment (save as PDF format).