|  |  |  |
| --- | --- | --- |
| Homework #1 | CS 111, Programming Fundamentals II  Central Washington University | **Computer Science** |

This is the first of 4 homework assignments. It is meant to review CS 110, as well as focus on our new material of more about claasses. Much of this homework should be review, from CS110. There are two programming tasks, each worth 30 points. Start **NOW**.

**Programming Task 1, CS 110, Review, 30 points**

We've all heard that the odds in gambling are always in the favor of the casino. But is it true? And what does that mean? For this programming task, you'll apply many of the concepts that you have learned in CS110, to “simulate” playing 1000 slot machines, and you'll tally how many of them result in a win.

If you are not familiar with slots, they are machines with three or more reels which spin when a button is pushed (Figure 1a). The reels have numbers or pictures of fruits written/drawn on them, and the goal is to have the reels spin, and stop, such that at least one of the 3 visible rows of numbers among the three reels are the same. For this homework problem, you will simulate a slot machine by writing a class, SlotMachine, that has 3 fields that are arrays of three integers (Figure 1b). Your class SlotMachine should have a method playMachine, which uses the Random class to generate random numbers and place them into the three arrays. The method isWinner will determine if any of the three arrays have three identical entries (a winning play is shown in figure 1c).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | 1 | 5 | 8 |  |  |  |  | | --- | --- | --- | | 3 | 3 | 4 |  |  |  |  | | --- | --- | --- | | 6 | 8 | 9 | | |  |  |  | | --- | --- | --- | | 2 | 4 | 9 |  |  |  |  | | --- | --- | --- | | 1 | 5 | 8 |  |  |  |  | | --- | --- | --- | | 4 | 4 | 4 | |
| 1a. A slot machine | 1b. Three arrays representing the three visible rows for three reels | 1c. Three arrays, the bottom of which is a “winner.” |

To start, create two java files, *SlotMachine.java* and *Play1000SlotMachines.java*, in the jGrasp or NetBeans IDE of your choice. SlotMachine is the class whose UML diagram is shown below. Each of the fields row1, row2, and row3 should be an array that contains three integers. The default constructor should invoke the playMachine method.

|  |
| --- |
| SlotMachine |
| - row1 : int[]  - row2 : int[]  - row3 : int[] |
| + SlotMachine()  + playMachine() : void  + isWinner() : boolean |

The file *Play1000SlotMachines.java* should contain a main method that creates an **array** of 1000 items, each an instantiated object of type SlotMachine. Use a for-loop in your main method to create each of the 1000 slot machines. Designate a counter variable to tally how many of the 1000 slot machines are “winners”. The last thing that your main method should do is print to the screen the number of winning slot machines. For example:

From 1000 slot machines, 35 were winners

This first programming task is meant to be a review. If you find yourself writing extensive code, then stop, and ask for help. The SlotMachine class file can be written so that it has fewer than 50 lines of code, and the file *Play1000SlotMachines.java* can be written in fewer than 20 lines of code (excluding comments).

Upload Play1000SlotMachines.java, SlotMachine.java and “Play1000SlotMachines output.docx” to Canvas.

Always put your name, date, and honor code( from syllabus-not what you think it is) on every file submitted.

Check the rubric posted on the course website to see a breakdown of the points for the first programming task.

**Programming Task 2, More about Classes, Static Class Members – 30 points**

Complete the first programming challenge, Area Class:

1. Area Class

Write a class that has three overloaded static methods for calculating the areas of the following geometric shapes:

• circles

• rectangles

• cylinders

Here are the formulas for calculating the area of the shapes.

Area of a circle: Area = πr**2**

where π is Math.PI and r is the circle's radius

Area of a rectangle: Area = Width X Length

Volume of a cylinder: volume= πr**2** h

where π is Math.PI, r is the radius of the cylinder's base, and

h is the cylinder's height

Area of a cylinder: Area= 2πr**2** + 2πrh

where π is Math.PI, r is the radius of the cylinder's base, and

h is the cylinder's height

Because the three methods are to be overloaded, they should each have the same name, but different parameter lists. Demonstrate the class in a complete program.

**In your main routine, you must invoke the static methods in the Area class. Do not create an instance of type Area.**

As with the first programming portion of this homework, this programming task shouldhelp you get back into the habit of coding, especially if it's been a while since you completed CS110.

Note that the class file can be written in fewer than 30 lines (excluding comments), and the java file with the main method can be written with fewer than 20 lines.

In your program where you use the methods of the Area class, also write to the screen the areas and volume that are computed using the static methods. For example:

The area of a circle with a radius of 20.0 is 1256.64

The area of a rectangle with a length of 10 and width of 20 is 200.00

The area of a cylinder with radius 10.0 and height 15.0 is 1570.8

Upload Area.java, AreaDriver.java and “AreaDriver, Area output.docx” to Canvas.

Always put your name, date, and honor code( from syllabus-not what you think it is) on every file submitted.

Check the rubric posted on the course Canvas to see a breakdown of the points that are possible for the first programming task.