PROGRAMMING ASSIGNMENT 1

Virtual Dice Throwing and Point-Circle-Cylinder Inheritance and Polymorphism

The first Programming II assignment has as target to get you to write two programs in order to enhance your knowledge of programming, to assimilate the taught material and also get to grips with object-oriented principles. It comprises two programs:

- a virtual dice throwing program; and
- a program exercising inheritance and polymorphism.

The first program should emulate virtual dice throwing. It should get from the standard input the number of sides of the virtual dice (should be >= 2) and the number of times to be thrown which should be a multiple of the number of sides, you should check for this and make sure it is the case. The program should then throw the dice and print out the number of times each side occurred. Given than random numbers will be used, you should be able to see that the higher the number of throws, the closer the results will be towards approximately equal occurrence of all sides (same probability). The program should deal gracefully with incorrect type of arguments through checks and exceptions. You will need to get the arguments in infinite loops "e.g. while (true)", continuing with "continue;" when an argument is wrong and finally breaking the loop with "break;" when the required argument has been correctly received. Note that while using Scanner to get these values, if a value is incorrect and causes an exception, the Scanner *nextLine* method should be used which advances to the next line of the standard input. Otherwise, the scanner keeps getting the same incorrect value and the program will keep looping.

Although obvious, it should be stated that this program does not exercise any object-oriented features per se, e.g. reusable object-oriented code, unlike the second one.

The second program has the purpose to get you to implement the Point, Circle and Cylinder classes as in the lecture notes in order to get to understand better, assimilate and exercise the inheritance features taught in the lectures. You should implement those three classes using inheritance, with Shape being an interface instead of an abstract class. You should pay special attention to reusing code from the class you inherit from so that you introduce the derived class with the minimal amount of code, as in the notes. You can find the code in the printed notes but not in the PDF versions on Moodle as the idea is to write the code by yourselves, realising how object-oriented features related to inheritance are exercised and understand / assimilate the relevant principles.

Having implemented those classes, you should write a ShapeTest program that gets input from the user to create a Point, Circle or Cylinder accordingly. The user should be able to create as many of these shapes and s/he wants. The menu should also allow the user to printout the created objects When the user finishes with passing relevant input, the program should print the created using the getName(), toString(), getArea() and getVolume() methods, in a similar fashion to the polymorphic behaviour test program in the notes. The program should deal gracefully with incorrect number and type of arguments through checks and exceptions.

Please note that correct indentation is *extremely* important and will affect the mark.

You should demonstrate the working programs and also the mean and variance program of assignment 0 to the lab helpers/myself in the lab session of February 14. You will be also asked to show the code and will be asked questions about the code.

You should also submit the MeanAndVariance.java from assignment 0 and the VDiceThrower.java and ShapeTest.java program from this assignment through moodle by the end of Sunday 16 February.