**CSE 220 Homework Assignment 2 (Due 4/8/22)**

1. **(30 pts)** Your first task deals with constructing a class hierarchy diagram. Recall that in class we looked at a simple example of a *Dictionary* class inheriting from a *Book* class. **Construct a class hierarchy diagram for varying types of books or any other media of your choice that satisfies the following criteria: there should be at least six classes ii. there should be at least three inheritance relationships** **depicted**. **Note: you are to construct a class hierarchy diagram -- not a UML Class diagram!** If you are unsure how to construct a class hierarchy diagram you should follow the example provided for Animals in the Chapter 9 Part 2 presentation. You do not need to specify variables or methods for the classes.



The diagram begins from the top with books being the most general descriptor. Below that, 3 different types of books all inherit from the books class. These 3 types of books are: Fiction, Non-Fiction, and Comic Books. From this we go down to the next level where each type of book has a specific book listed that falls into that category, these specific books all inherit from their book type which inherits from the Book class at the top of our diagram.

**2) (30 pts)** Your second task deals with inheritance and *overriding methods*.   
a. Explain the difference between method overloading and method overriding in object oriented programming.

Method Overloading is a compile-time polymorphism. With OOP, method overloading is when more than one method shares the same method name with different signatures in the class. The return type in these methods can be the same but it is not required to be, but the parameters will need to be changed if the return is different. Overloading helps increase the readability of a program.

Method Overriding is a run-time polymorphism. With OOP, method overriding is when methods will have the same name and signature. The return type must be the same as the method it is relating too. Overriding is used to grant specific implementation of the method which is already provided by its parent class or its subclass. Method Overriding always needs inheritance relationships.

b. Can constructors be i. overloaded and/or ii. overridden in Java? Explain your answers.

Constructors cannot be overridden in Java. This would cause a compile run time error due to the user writing a super class’s constructor in the sub class would cause the compiler to treat it as a method. But there is no return type and an error occurs.

c. Explain the function of the *super* keyword in Java.

The super keyword in Java refers to the superclass objects and is a primary concept used in Inheritance. It is used to call on super class methods, have access to the super class constructor, and even refer to parent class instance variables.

3) You are to write three hierarchical classes in Java with the specifications below.

a. The **Polygon** class consists of the following:  
i. an array of doubles called **edges**, which represent the edges/sides of the polygon: edges should be accessible to inheriting classes, but not public.  
ii. a constructor that takes in an array of doubles and uses it to initialize edges  
iii. a method **double perimeter()** that returns the polygon’s perimeter

b. A **RightTriangle** inherits from Polygon and represents a right triangle with the first two edges representing the legs, and the third representing the hypotenuse. It has the following additions/changes:  
i. A constructor that must verify that the input array **edges** is of size three, and that the third edge (approximately) satisfies the Pythagorean theorem (a2+b2=c2). If either condition is failed, an exception must be thrown.   
ii. A method **double area()** that returns the area of the right triangle

c. A **Rectangle** inherits from Polygon and represents a quadrilateral whose first and third edges have equal length, and whose second and fourth edges have equal length. It has the following additions/changes:  
i. A constructor that must verify that the input array **edges** is of size four, and that edges 1 and 3 match, as well as that edges 2 and 4 match. If any of these three conditions is failed, an exception must be thrown.   
ii. A method **double area()** that returns the area of the rectangle.

*Reminders regarding exceptions:  
i. A constructor that throws an uncaught exception must indicate this explicitly. For example, your signature for the RightTriangle constructor should likely be:****public RightTriangle(double[] edges) throws Exception {…}****ii. You can throw the exception using syntax* ***throw new Exception(<message>);***

**Responses to Problems #1 and #2 should be in .doc(x) or .pdf format. Upload a .zip file containing this file and your .java files for problems 3 to Blackboard as your assignment submission.**

**Your submitted file should have the filename “LN\_FN\_4.zip” where LN is your last name and FN is your first name.   
*Remember: do not submit your .bluej files: they have no source code!***