CIST 2371 Introduction to Java

Unit 06 Lab
Due Date: _____

All files will go into a **Unit06** folder.

Part 1 – Inheritance and Polymorphism

For this part you will design five classes: Person, Student, Employee, Faculty and Staff. A Person class has two subclasses: Student and Employee. The Employee class has two subclasses, Faculty and Staff. A Person has a name, address, phone number and email address. A student as a class-status (freshman, sophomore, junior or senior). An Employee has an office, salary and date hired. A faculty member has office hours and rank. A staff member has a title. All these attributes are Strings, except salary which is a double. Override the toString() method in each class to return a string containing the class name and the person's name.

You will be turning in the following items:

1. Create a UML class diagram that illustrates the design. This will be done by using UMLet and MS Word or some other word processor (export the drawing from UMLet and import the image file into a word processing document). he document should have your name, course assignment and date at the top like this:

   Sally Doe
   CIS 2371 Summer 2014
   Unit 06 Lab Part 1
   June 12, 2014

   The file should be an PDF formatted file and be named Unit06Part1_<first initial><lastname>.pdf, for example **Unit06Part1_sdoe.pdf**. Any word processor can save or export to PDF.
2. Implement each class in the class diagram: **Person.java, Student.java**, etc.
3. Create a test program called **Unit06_Prog1.java** that creates an instance of each of the classes above, sets the data and calls their toString() method and prints out the result to standard out. Here is a sample run:

Part 2 – Abstract Classes and Interfaces

For this part you will design two classes and an interface; the GeometricObject class, the Square class and the Colorable interface. GeometricObject has the attributes of color (String), filled (Boolean) and creationDate (String). The Colorable inteface defines one method: howToColor()

which returns a String and takes no arguments. The Square class has the attribute side (a double) which is the length of its side. It also has the methods getArea() and getPerimeter() which return doubles. All classes have thier getters and setters. The Square class extends the GeometricObject class and implements the Colorable interface. The implementation of the howToColor() method returns the String "color all four sides".

You will turn in the following

1. Create a UML class diagram that illustrates the design. This will be done by using UMLet and MS Word or some other word processor (export the drawing from UMLet and import the image file into a word processing document). You must put your name, the name of the class the name of the exercise and the date at the top left of your paper as stated in part 1. The file should be an PDF formatted file and be named Unit06Part2_<first initial><lastname>.pdf, for example **Unit06Part2_sdoe.pdf**.

2. Implement each class and interface; **GeometricObject.java**, **Colorable.java** and **Square.java**

3. Create a test program called **Unit06_Prog2.java** that creates an instance of the Square class and calls it's howToColor() method, printing out the String on standard out.

What To Turn In
You will zip up the **Unit06** folder containing your source code files and your UML diagrams into a zip file named Unit06_<your first initial><your lastname>.zip. For example Tom Swift's zip file would look like this: **Unit06_tswift.zip**

Be sure to test this file before turning it in. Copy it into a temp folder, unzip it and try to compile and run the files in that temp folder. Once you are satisfied that your zip file is OK, turn it in via the ANGEL drop box for this Unit.

The Comment Block
EVERY source code file you turn in for this course must have this comment block (customized for the situation) or you will get no credit (kind of like forgetting to put your name on your term paper – bad move). Use the same comment block you have used to date, adjusted for each program.

Rubric (50 points total)

P1 UML Diagram – PDF document with header & correct diagram, all 5 classes: 5 of 5pts
P1 All classes have comment block: 2 of 2pt
P1 All classes have correct attributes: 2 of 2pts
P1 All classes have getters and setters: 4 of 4 pts
P1 Unit06_Prog1 operates correctly: 10 of 10pts

P2 UML Diagram – PDF document with header & correct diagram, 2 classes and interface: 5 of 5pts
p2 All source code files have the comment block: 4 of 4 pts
P2 Colorable Inteface has correct method: 3 of 3 pts
P2 Square class extends GeometricObject and implements Colorable: 5 of 5pts
P2 Unit06_Prog2 tests howToColor on the square class and it works correctly: 10 of 10pts