Assignment\_2\_Java\_Inheritance

Due date: 10/26 2016 Wednesday

Course: NJIT Java, Android, Data structures IT114

Instructor: ilker Kiris

Student: <your Name and Last Name goes here>

**SETUP and how to submit**

1. Create Java Project (highly suggesting to create it in STS) and put the java packages into a source directory called src
2. Add README.md file at root (top) level of project explaining what is happening
3. Create a class diagram (highly suggesting to install ObjectAid eclipse plugin to STS and using that to create class diagram and an image of the class diagram and put them to directory called *design* under root level)
4. Create github account, create an organization called it114-assignments-yournameLastname
5. Create a github repository called assignment-java-inheritance-yournameLastname in above organization
6. Create a git repository of your project, commit your code (first add to index then commit) to local repository, then set above created github repository as remote repository for the project and finally push to remote github repository
7. Email ilker the questions and your answers. And link to your above created github repository.

**CODING**

Create a Java project with classes depicted in below class diagram, that;

- create interfaces (Person, Studying),

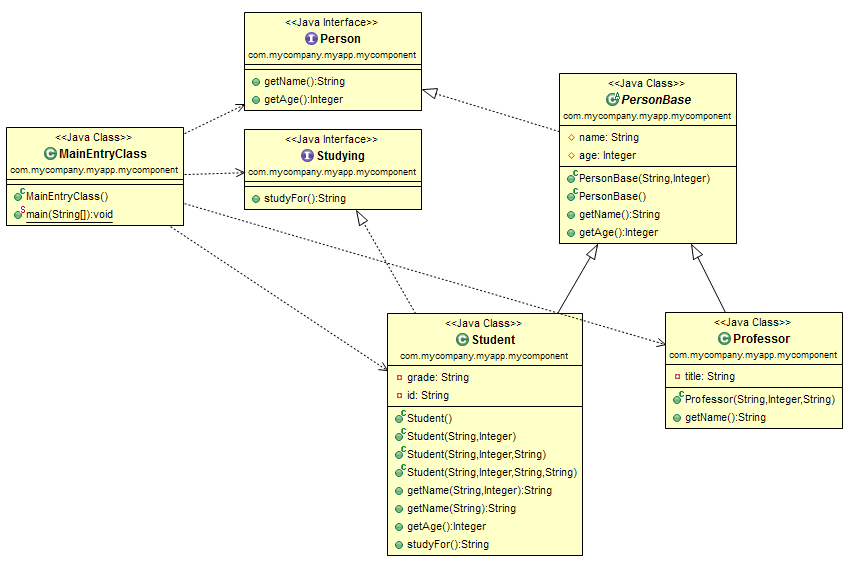
- create abstract base class that contain common attribute and methods of classes that implement Person interface (PersonBase),

- create multiple implementation classes of Person interface (Professor, Student),

- create class (Student) implementing multiple interfaces (Student, Person),

- use polymorphism in MainEntryClass, in which left hand side variables will be of type interfaces (Person, Studying) while they point to real implementation (Professor, Student)

**Example class diagram**;



**Questions**;

1. What are interfaces for?

**An interface is a reference type in Java. It’s similar to class. It is a collection of abstract methods.**

1. What are abstract classes for?

**Abstract classes are classes that contain one or more abstract methods.**

1. What are classes for?

**A class is a blueprint or a template for creating different objects which defines its properties and behaviors.**

1. What is the relationship between interfaces and classes?

**Interface promotes multiple inheritances so it is best for defining the type. A Class implements multiple interfaces but Interface extends multiple interfaces, So the interface is more polymorphic**

1. How can almost emulate multiple inheritance in Java?
2. What is method overloading?

**Method Overloading is a feature that allows a class to have two or more methods having same name, if their argument lists are different.**

1. What is method overwriting?

**Overriding means to override the functionality of an existing (inherited) method.**

1. Why polymorphism together with interfaces important in Java?

**Because they allow objects to have a more dynamic form to further enhance the code.**

1. Can you create an instance of Abstract class?

**No: abstract classes cannot be instantiated**

1. What things form a method’s full signature?

**Access modifier, method complete with brackets, semi-colon**

1. How many access levels are there? What are they? What they mean?

There are 4:

**Private**: Only the class in which it is declared can see it

**Protected**: can be seen by child or package members

**Public**: Can be seen by everyone

**Default**: when no access level is declared, Java uses default

1. What do these access levels mean for inheritance?
2. When a child class is overwriting a method from its parent, can it make the method more or less accessible? Or does it have to keep access level same?

**No: It must keep the access level the same.**