CS 2334: Lab 6
Abstract Classes & Interfaces

Abstract Class

Few important points to remember about abstract classes:

- An abstract class is a class that is declared '***abstract***'
  - It can, but does not have to, include abstract methods
- Abstract classes can not be instantiated but *they* can be subclassed using the ***extends*** keyword
- An abstract method is a method that is declared without an implementation
  - Requires the ***abstract*** keyword

Abstract Class

- If a class includes abstract methods, then the class itself must be declared as abstract.

```
//declaring class abstract
public abstract class Person{
        //declaring method abstract
        abstract void generateID( );
}
```

- When a child class extends an abstract class, it must either:
    - Provide implementations for all abstract methods from its parent class, or
    - Also be abstract
- Child classes can reference the constructor of abstract class by using super()

Interface

- Interface is a blueprint of a class.
  - It has static constants and abstract method only
- It can not be instantiated
- Interface represents is-a relationship

```
interface printable{
            void print();
}

class print implements printable{
            public void print(){System.out.println(“Hello”);}
}
```

What is the Difference between Abstract Classes and Interfaces?

Abstract Classes vs Interfaces

- Abstract class can have abstract and non-abstract methods while interface can only have abstract methods
- Class inheritance (extension) does not support multiple inheritance, while a class can implement an arbitrary number of interfaces
- Abstract classes can have final, non-final, static or non-static variables while interface can have only final and static variables

Lab 6: Representing Shapes

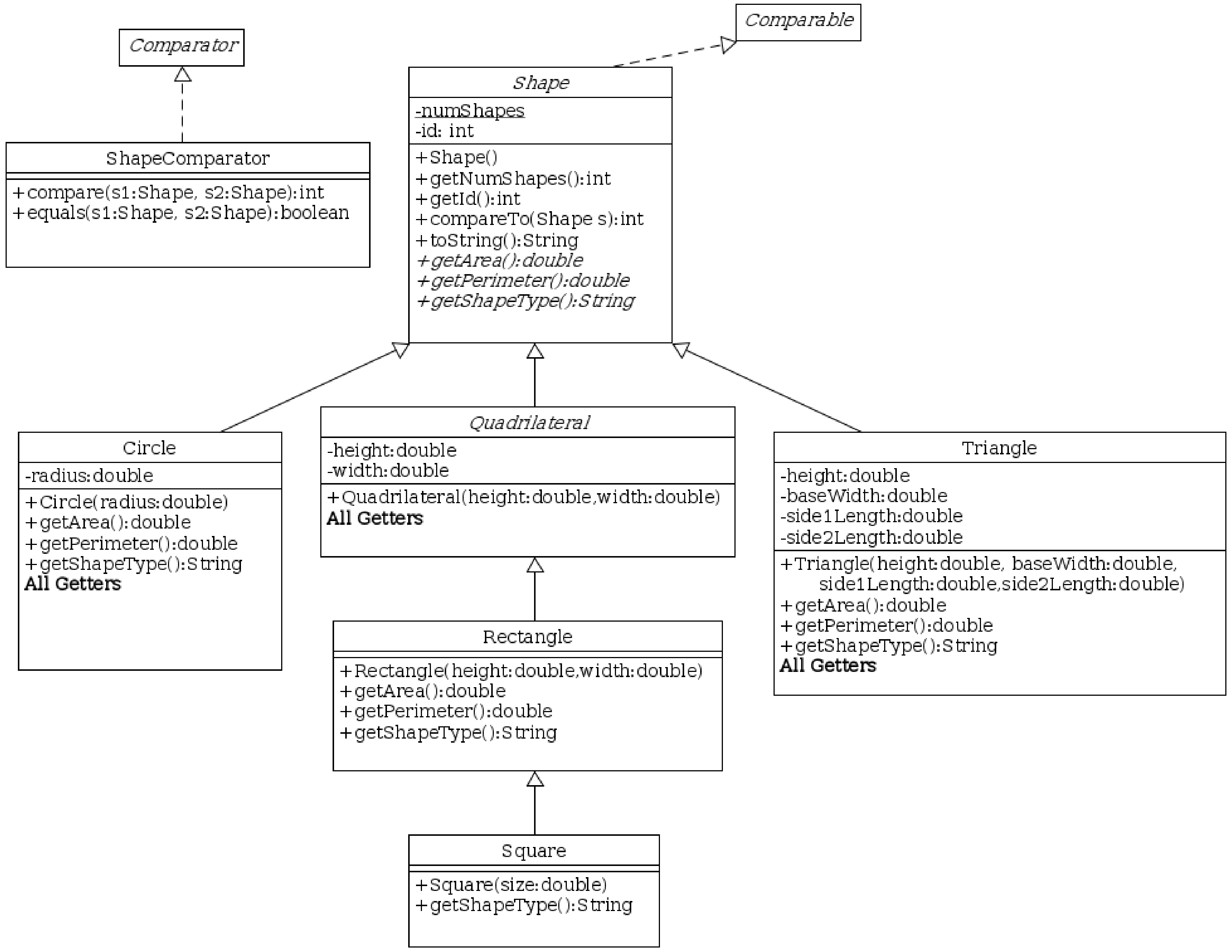
Given a UML diagram that describes the relationships between shape classes:

- Implement each class, including the specified instance variables and methods
- Implement testing procedures for the classes

Representing Different Geometrical Shapes

Interfaces from the Java API:

- Comparable
- Comparator



Lab 6 Preparation

- Download lab6-initial.zip
- Import into your Eclipse environment

(details of how to do this are in the lab specification)

Lab 6 Notes

- Create each class in the UML diagram
  - Include all methods and instance variables, with the specified visibility
  - Watch spelling and casing
  - Use the default package
- Implement attributes and methods
  - Classes are dependent on each other, so you can have temporary errors while you implement
- Expand on the given test class to make sure it all works

Submission

- Submit only one file: lab6.zip (casing matters)
- Due date: Friday, October 2nd @11:59pm
- Submit to lab6 dropbox on D2L