**CSCI 2302**

**Inheritance & Polymorphism Chapter**

**Inheritance Lab with GeometricObject & Square Classes**

Contents

[Intro 1](#_Toc84359590)

[Learning Goals 1](#_Toc84359591)

[Notes 1](#_Toc84359592)

[Task 6](#_Toc84359593)

[Submit 8](#_Toc84359594)

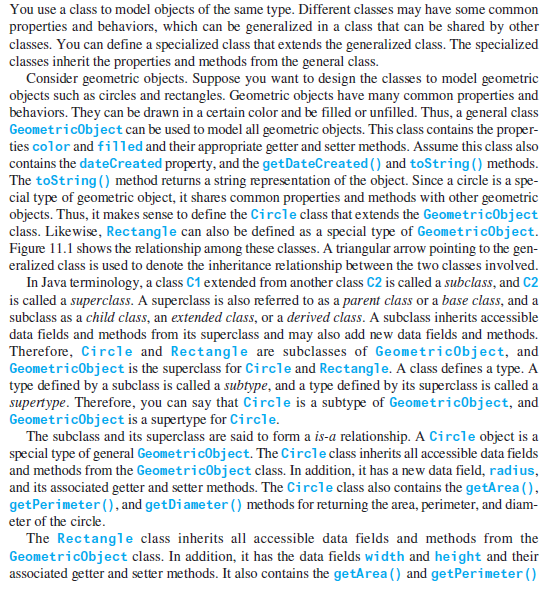
Intro: Inheritance is one of the most important aspects of Object-Oriented Programming (OOP). The key to understanding Inheritance is that it provides code re-usability. In place of writing the same code, again and again, we can simply inherit the properties of one class into the other. OOP is all about real-world objects and inheritance is a way of representing real-world relationships.

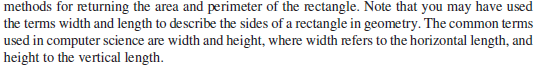
Inheritance is the procedure in which one class inherits the attributes and methods of another class. The class whose properties/states/fields and methods are inherited is known as the Parent class. And the class that inherits the properties/states/fields from the parent class is the Child class; thus, the child class has the parent’s properties/states/fields and methods and it has its own properties/states/fields and methods.

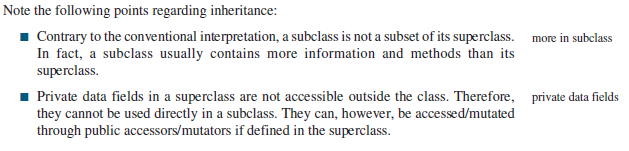
Inheritance avoids redundancy, makes the program easy to comprehend and easy to maintain.

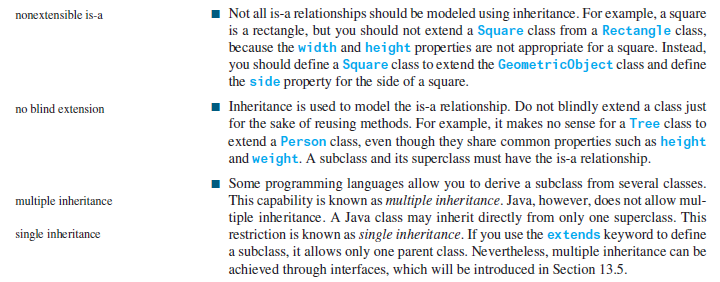
Learning Goals: To demonstrate how inheritance is implemented in a Java program. To compare how this class relationship, is-a relationship, differs with aggregation, has-a relationship. So that in the future we can select which class relationship is needed as we build our own program systems.

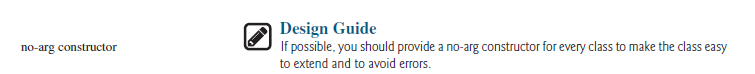
Notes:

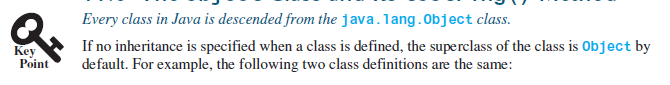








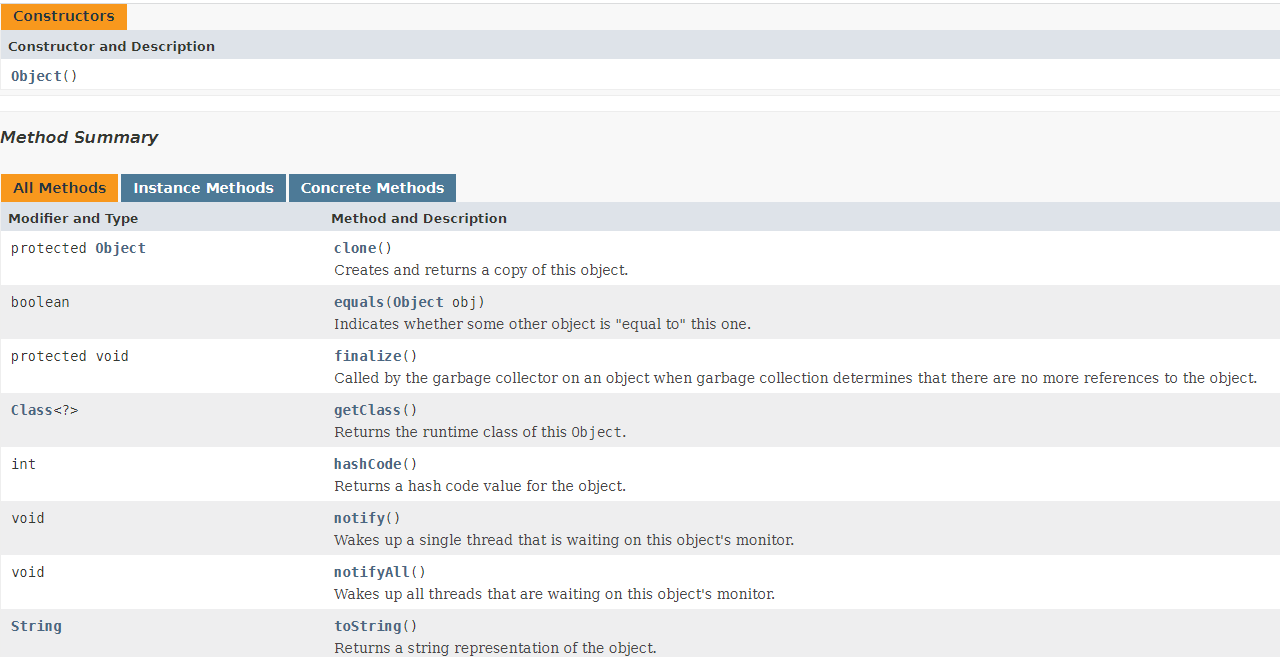


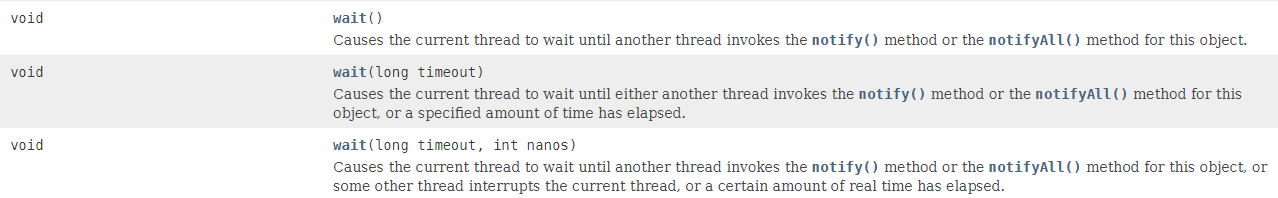


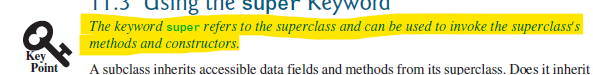
Every time you define a class, it is inheriting from the Object class that is pre-defined in Java. This means that it has everything that the Object class has.

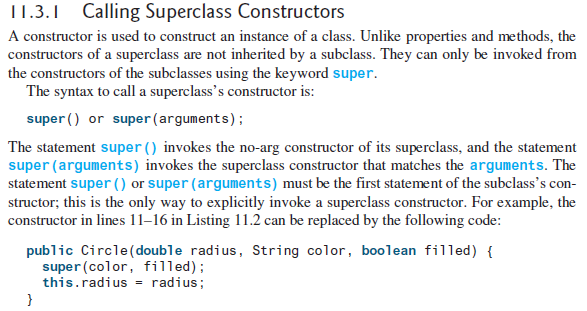


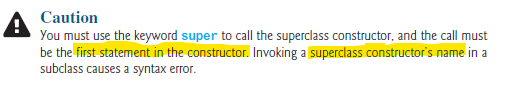
https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html

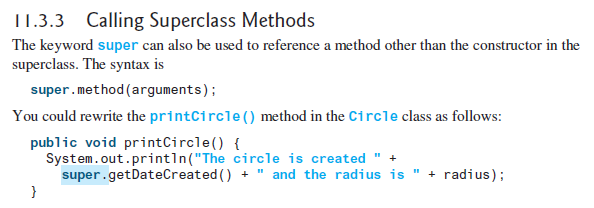


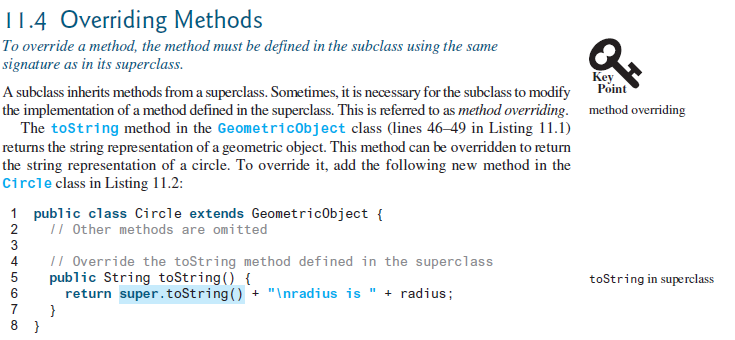


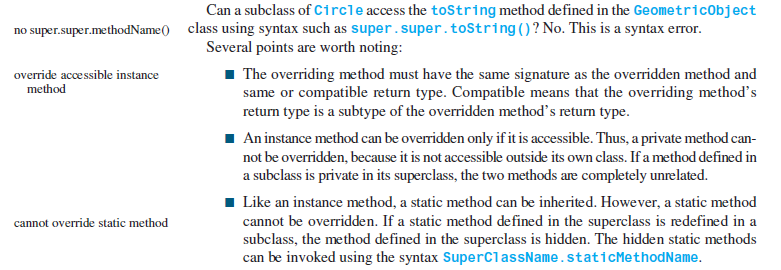


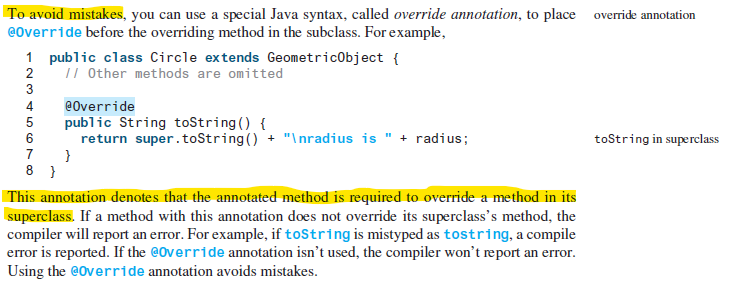


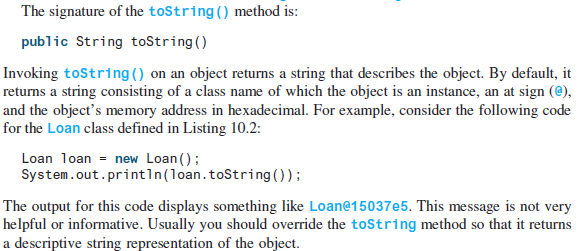






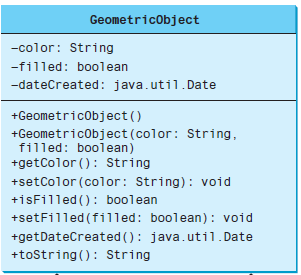






Task: Complete the steps outlined below in a file named mysfasuUsername\_InheritanceLab.java.

1. Download the GeometricObject.java file, save that file where mysfasuUsername\_InheritanceLab is/will be located on your computer.
2. Define/Implement a Square class based on the following UML diagram.



|  |
| --- |
| Square |
| -side: double |
| +Square()  +Square(side: double)  + Square(color: String, filled: boolean, side: double)  +setSide(side: double): void  +getSide(): double  +anotherMethodToGetDateCreated(): String  +toString(): String |

1. In the mysfasuUsername\_InheritanceLab, in the main method, instantiate:
   1. a GeometricObject object
   2. A Square object with no-args
   3. A Square object with a side value of 2
   4. A Square object with color = purple, filled = true, side = 3
   5. An Object object
2. For each instantiated object from step 2, invoke the toString method.
3. Invoke the anotherMethodToGetDateCreated method using the Square objects.

Submit: Submit your mysfasuUsername\_InheritanceLab.java (make sure to include the Square class) file in the Dropbox in Brightspace by D2L.