1. Define the abstract **GeometricObject** as the base class for all classes like Circle, Rectangle, Triangle… defined earlier. It should have abstract method getArea and getPerimeter. Define a generic method costOfPaintingashape(GeometricObject g) and use to find cost for different geometric objects placed in an array based on the User.
2. (*Enabling* **GeometricObject** *comparable*) Modify the **GeometricObject** class to implement the **Comparable** interface, and define a static **max** method in the **GeometricObject** class for finding the larger of two **GeometricObject** objects. Draw the UML diagram and implement the new **GeometricObject** class. Write a test program that uses the **max** method to find the larger of two circles and the larger of two rectangles.
3. (*The* **Colorable** *interface*) Design an interface named **Colorable** with a **void** method named **howToColor()**. Every class of a colorable object must implement the **Colorable** interface. Design a class named **Square** that extends **GeometricObject** and implements **Colorable**. Implement **howToColor** to display a message **"Color all four sides"**. Draw a UML diagram that involves **Colorable**, **Square**, and **GeometricObject**.
4. Write a test program that creates an array of five **GeometricObject**s. For each object in the array, invoke its **howToColor** method if it is colorable.
5. (*Finding the largest object*) Write a method that returns the largest object in an array of objects. The method signature is:
   public static Object max(Comparable[] a)
   All the objects *are* instances of the **Comparable** interface. The order of the objects in the array is determined using the **compareTo** method. Write a test program that creates an array of ten strings, an array of ten integers, and an array of ten dates, and finds the largest string, integer, and date in the arrays.