Write a Java application that meets the following specifications.

There are two types of Pizza, namely NeapolitanPizza and SicilianPizza. NeapolitanPizzas are round, while SicilianPizzas are rectangular. NeapolitanPizzas weigh one ounce per square inch, while SicilianPizzas weigh two ounces per square inch. Every Pizza initially has eight slices with equal areas.

Each Pizza contains zero or more toppings, stored in a list. The *only* available toppings are anchovies, pepperoni, and mushrooms.

Every Pizza has the following methods. At least one of these methods should be abstract in Pizza, but I have not indicated which one(s):

public void addTopping(Topping t)	adds a topping to the pizza
public boolean checkTopping(Topping t)	returns true if the pizza has a particular topping, otherwise false
public Double getCurrentArea()	returns the current surface area
public double getCurrentWeight()	returns the current weight of the pizza
public double eatSlice()	returns the area of a slice if one is available; otherwise returns zero. Also does whatever is necessary to ensure that future calls to getCurrentArea() and getCurrentWeight() will reflect the reduced amount of pizza remaining.

Pizza also implements Comparable<Pizza> in a way that will result in a sort order in ascending order by current area.

Each type of Pizza needs a constructor. NeapolitanPizza's constructor should take the radius as a parameter, while SicilianPizza's constructor should take the length and width.

Hints:
Remember that variables that are protected, rather than private, in Pizza can be accessed from the subclasses.
You will need to import the Topping to the JUnit test case.

Code Pizza, NeapolitanPizza, and SicilianPizza.