public class Pair<T>

{

private T first;

private T second;

public Pair() { first = null; second = null; }

public Pair(T first, T second) {  
 this.first = first;

this.second = second;

}

public T getFirst() { return first; }

public T getSecond() { return second; }

public void setFirst(T newValue) { first = newValue; }

public void setSecond(T newValue) { second = newValue; }

}  
  
**Bounds for Type Variables  
-** public static <T extends Comparable> T min(T[] a)

**Type Parameters Cannot Be Instantiated with Primitive Types  
You Cannot Create Arrays of Parameterized Types**

**If you need to collect parameterized type objects, simply use an ArrayList:ArrayList<Pair<String>> is safe and effective.**

You Cannot Instantiate Type Variables  
You Cannot Construct a Generic Array  
Type Variables Are Not Valid in Static Contexts of Generic Classes  
You Cannot Throw or Catch Instances of a Generic Class  
  
**You Can Defeat Checked Exception Checking**@SuppressWarnings("unchecked")

**Inheritance Rules for Generic Types**



**The Wildcard Concept**  
Pair<? extends Employee>  
  
**Supertype Bounds for Wildcards**  
void setFirst(? super Manager)  
  
