```java
//Generics_Intro.java
//import java.util.*;
public class Generics_Intro {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Integer[] myInt = {5, 4, 6, 10, 12};
		Character [] myChar = {'y', 'a', 'b', 'q', 'u'};

		printMe(myInt);
		printMe(myChar);
		// This is a problem -- it won't work
	}


	/*
	public static <E> void printMe(E[] myIntArray) {
		for (int i=0; i<myIntArray.length; i++)
			System.out.printf(myIntArray[i] + " ");
		System.out.println();
	}
	*/



	public static void printMe(Integer[] myIntArray) {
		for (int i=0; i<myIntArray.length; i++)
			System.out.printf(myIntArray[i] + " ");
		System.out.println();
	}


	public static void printMe(Character[] myIntArray) {
		for (int i=0; i<myIntArray.length; i++)
			System.out.printf(myIntArray[i] + " ");
		System.out.println();

	}
}
```

```java
//Generic_Return.java
// A generic can be a subtype of another type. This is called bounded
public class Generic_Return {

	public static void main(String[] args) {
		System.out.println("The maximum of 3, 5, 1 is " + Find_Maximum(3, 5,
1));
		System.out.println("The maximum of b, c, d is " + Find_Maximum('c', 'b',
'd'));

	}

	public static <T extends Comparable<T>> T Find_Maximum (T a,T b, T c) {
		T m_value = a;

		if (b.compareTo(a) > 0)
			m_value = b;

		if (c.compareTo(m_value) > 0)
			m_value = c;

		return m_value;
	}

}
```

```java
//TestArrayList.java


import java.util.ArrayList;
public class TestArrayList {

	public static void main(String[] args) {
		ArrayList<String> myNames = new ArrayList<>();
		myNames.add("Thomas");
		myNames.add("Jessica");
		myNames.add("Michael");
		myNames.add("Ted");
		// myNames.add(3); // gives you a compilation error

		System.out.println("Number of names is: " + myNames.size());
		System.out.println("The location of Jessica is: " +
myNames.indexOf("Jessica"));
		System.out.println("The location of Thomas is: " +
myNames.indexOf("Thomas"));

		System.out.println("Is Ted available in my list?");
		boolean flag = myNames.contains("Ted");

		if (flag == true)
			System.out.println("Yes. Ted is here");
		else
			System.out.println("No. Ted is not here");


		System.out.println("Items in my list before any changes:");
		System.out.println(myNames.toString());

		//Insert a new name at index 1
		myNames.add(1, "Alex");

		System.out.println("Items in my list after adding Alex:");
		System.out.println(myNames.toString());

		myNames.remove("Ted");
		System.out.println("Items in my list after removing Ted:");
		System.out.println(myNames.toString());


		// Printing the first location
		System.out.println(myNames.get(0));
		System.out.println(myNames.get(1));


		ArrayList<Integer> myIntegers = new ArrayList<>();
		myIntegers.add(10);
		myIntegers.add(20);
		myIntegers.add(6);
		myIntegers.add(3);

		System.out.println(myIntegers.get(0));
```

```
            ArrayList<Double> myDouble = new ArrayList<>();
            myDouble.add(5.5);
            myDouble.add(3.2);
            Double doubleObject = myDouble.get(0);
            double d = doubleObject;

            System.out.println(d);

        }

}
```

```java
// How to define a generic class?
//AddingElements.java
public class AddingElements<E> {
          private E element;

          public void addElement(E t) {
             element = t;
          }

          public E get() {
             return element;
          }

          public static void main(String[] args) {
                AddingElements<Integer> integerBox = new AddingElements<Integer>();
                AddingElements<String> stringBox = new AddingElements<String>();

             integerBox.addElement(new Integer(10));
             stringBox.addElement(new String("Hello World"));

             System.out.println("My Integer Element: " +  integerBox.get());
             System.out.println("My String Element: " + stringBox.get());
          }
       }
```