Core Java Day-5

**Q.** import java.util.\*;  
class Test {  
public static void main(String[] args) {  
// insert code here  
obj.add("one");  
obj.add("two");  
obj.add("TWO");  
System.out.println(x.poll());  
}  
}  
  
Which, inserted at // insert code here, will compile?

**1)**. List<String> obj = new LinkedList<String>();

**2)**. **TreeSet<String> obj = new TreeSet<String>();**

**3)**. HashSet<String> obj = new HashSet<String>();

**4)**. **Queue<String> obj= new PriorityQueue<String>();**

**Q.** class GenericsDemo<T>  
{  
 T data;  
 public GenericsDemo(T data)  
 {  
  this.data = data;  
 }  
}Which of the given statement is true about above code?

**1)**. GenericsDemo object can be created by passing any type of parameter

**2)**. GenericsDemo object can be created as given below  
GenericsDemo<String>obj = new GenericsDemo<String>();

**3)**. **GenericsDemo object can be created as given below  
GenericsDemo<String>obj = new GenericsDemo<String>("xyz");**

**4)**. Can not create object of given class

**Q.** class CreateObject  
{  
public static void main(String []args)  
{  
Set set = new TreeSet();  
set.add("Priya");  
set.add("Ritu");  
set.add(100);  
}  
}  
  
What will happen after the execution of above code?

**1)**. All elements will be successfully added to set

**2)**. last element 100 will not be added to set

**3)**. **ClassCastException**

**4)**. IlleagalStateException

**Q.** Which of the following methods are declared in List interface ?

**1)**. **get(int index)**

**2)**. **iterator()**

**3)**. listIterator()

**4)**. **remove()**

**Q.**Which of the given method must be overridden by a class,after implementing Comapartor?

**1)**. int compare(Object obj);

**2)**. int compareTo(Object obj);

**3)**. int compareTo(Object obj1,Object obj2);

**4)**. **int compare(Object obj1,Object obj2);**

**Q.**A programmer has an algorithm that requires a java.util.List that provides an efficient   
implementation of add(0,object), but does NOT need to support quick random access.   
What supports these requirements?

**1)**. ArrayList

**2)**. Queue

**3)**. Linear List

**4)**. **LinkedList**

**Q.** What is the output of the following?           
  
import java.util.\*;  
public class Test {  
public static void main(String[] args)   
{  
 Set set = new TreeSet();  
 set.add("anu");  
 set.add("anil");  
 set.add("sunil");  
 for(Object str:set){  
 System.out.print(str + " ");  
}  
}  
}

**1)**. **anil anu sunil**

**2)**. anu anil sunil

**3)**. Compilation error

**4)**. Exception

**Q.**A \_\_\_\_\_\_\_\_\_\_ is used to walk through a collection and can remove elements from the collection during the iteration.

**1)**. Enumeration

**2)**. **Iterator**

**3)**. ArrayList

**4)**. Vector

**Q.**Which collection class allows you to grow or shrink its size and provides indexed access to its elements, but whose methods are not synchronized?

**1)**. java.util.HashSet

**2)**. java.util.Vector

**3)**. **java.util.ArrayList**

**4)**. java.util.List

**Q.** class DemoCmp //line 1  
{  
int number;  
public DemoCmp(int num)  
{  
number=num;  
}  
//line 2  
}  
public class CreateDemo   
{  
public static void main(String []args){  
TreeSet<DemoCmp>set = new TreeSet<DemoCmp>();  
set.add(new Demo(8));  
set.add(new Demo(2));  
set.add(new Demo(3));  
}}  
Which code need to be inserted at line 1 and line 2  
for successful execution of above code?

**1)**. **//line 1  
class DemoCmp implements Comparable  
  
//line 2  
public int compareTo(Object obj)  
{  
return number-obj.number;  
}**

**2)**. //line 1  
class DemoCmp implements Comparator  
  
//line 2  
public int compare(Object obj1,Object obj2)  
{  
return obj1.number-obj2.number;  
}

**3)**. //line 1  
class DemoCmp implements Comparable  
  
//line 2  
public int compare(Object obj1,Object obj2)  
{  
return obj1.number-obj2.number;  
}

**4)**. //line 1  
class DemoCmp implements Comparator  
  
//line 2  
public int compareTo(Object obj)  
{  
return number-obj1.number;  
}