Queue Interface

- The implementation classes for Queue Interface are LinkedList & PriorityQueue.
- In Queue elements are stored in FIFO order.
- If we are creating an object for LinkedList with LinkedList reference variable, then we can access complete functionality of Queue & List.
- From 1.5v onwards LinkedList also implements Queue interface.

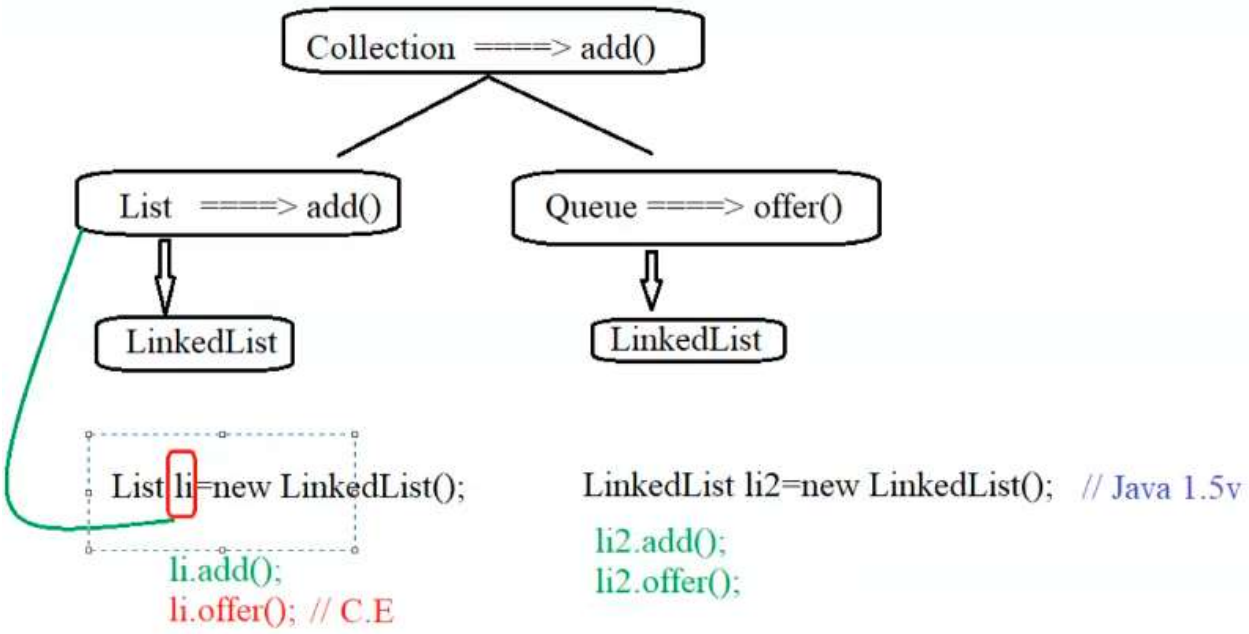
Queue Interface Methods

Method	Description
Offer(Object o);	Add an element in to Queue
Object poll() ;	To remove and return first element of the Queue (returns null if the queue is empty)
Object remove();	To remove and return first element of the Queue (NoSuchElementException when the queue is empty)
Object peek();	To return first element of the Queue without removing it

PriorityQueue

- It doesn't maintains insertion order and returns the elements in ascending order (Smallest Number first).
- In PriorityQueue the top element is always the smallest element.
- It doesn't accept null.
- PriorityQueue is available since jdk1.5V.
- It allows duplicate values, default capacity is 11.

```
PriorityQueue q=new PriorityQueue();
PriorityQueue q=new PriorityQueue(int initialcapacity);
```

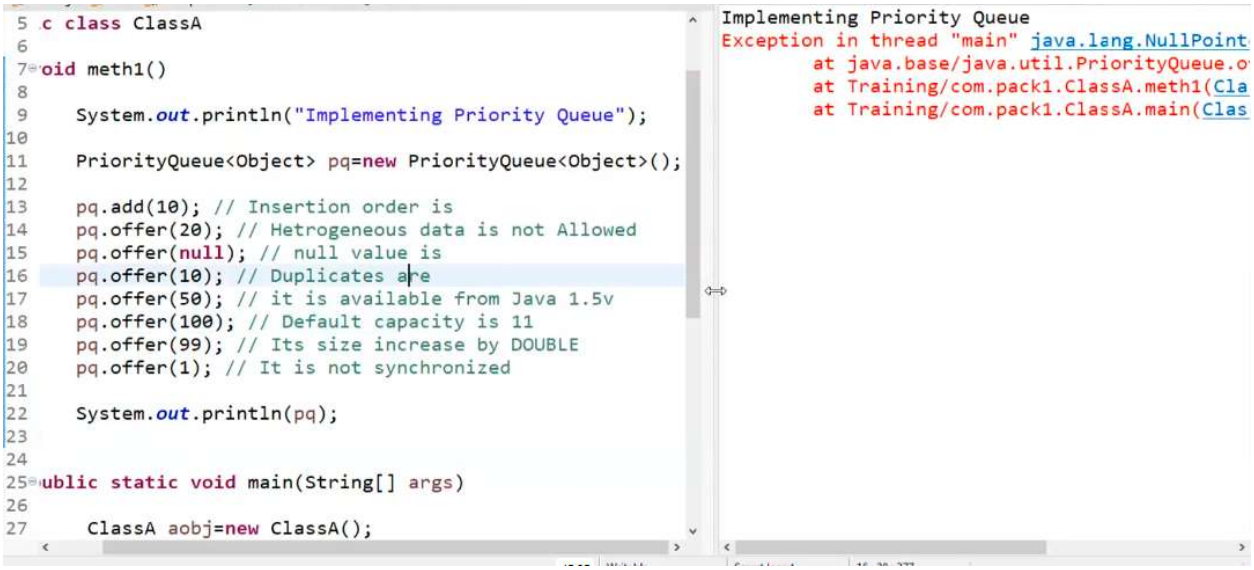
	A	B	C	D	E	F	G	H
1		ArrayList	Vector	LinkedList	Hashset	LinkedHashset	Treeset	PriorityQueue
2	Insertion Order	Maintained	Maintained	Maintained	Not Maitained	Maintained	Not Maitained (Sorting order==> Asc)	Not Maitained (The first
3	Duplicate Elements	Allowed	Allowed	Allowed	Not Allowed	Not Allowed	Not Allowed	Allowed
4	Null Value	Allowed	Allowed	Allowed	Allowed	Allowed	Not Allowed	Not Allowed
5	Default Capacity	10 (increases byHalf)	10 (increases by double)	0	16 (Load factor:0.75)	16 (Load factor:0.75)	16 (Load factor:0.75)	11
6	Hetergeneous elements	Allowed	Allowed	Allowed	Allowed	Allowed	Not Allowed	Not Allowed
7	Available from	1.2v	1.0v (Legacy Class)	1.2v	1.2v	1.4v	1.2v	1.5v
8	Synchronization	NOT	YES	NOT	NOT	NOT	NOT	NOT

```java
import java.util.LinkedList;
import java.util.List;

public class ClassA
{
    void meth1()
    {
        System.out.println("meth1() called\n");

        List <Object>ll1=new LinkedList<Object>(); // 1st Object

        LinkedList<Object> ll2=new LinkedList<Object>(); // 2nd Object

        ll1.add(10);
        ll1.offer(20);

        ll2.add(100);
        ll2.offer(200);
    }
}
```

```java
public class ClassA
{
    void meth1()
    {
        System.out.println("Implementing Priority Queue");

        PriorityQueue<Object> pq=new PriorityQueue<Objec

        pq.add(10); // Insertion order is
        pq.offer("Java"); // Hetrogeneous data is
        pq.offer(null); // null value is
        pq.offer(10); // Duplicates are
        pq.offer('A'); // it is available from Java 1.5v
        pq.offer(true); // Default capacity is 11
        pq.offer(99); // Its size increase by DOUBLE
        pq.offer(1); // It is not synchronized

        System.out.println(pq);
    }

    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
```

```
Implementing Priority Queue
Exception in thread "main" java.lang.ClassCast
        at java.base/java.lang.String.compareT
        at java.base/java.util.PriorityQueue.s
        at java.base/java.util.PriorityQueue.s
        at java.base/java.util.PriorityQueue.o
        at Training/com.pack1.ClassA.meth1(Cla
        at Training/com.pack1.ClassA.main(Clas
```

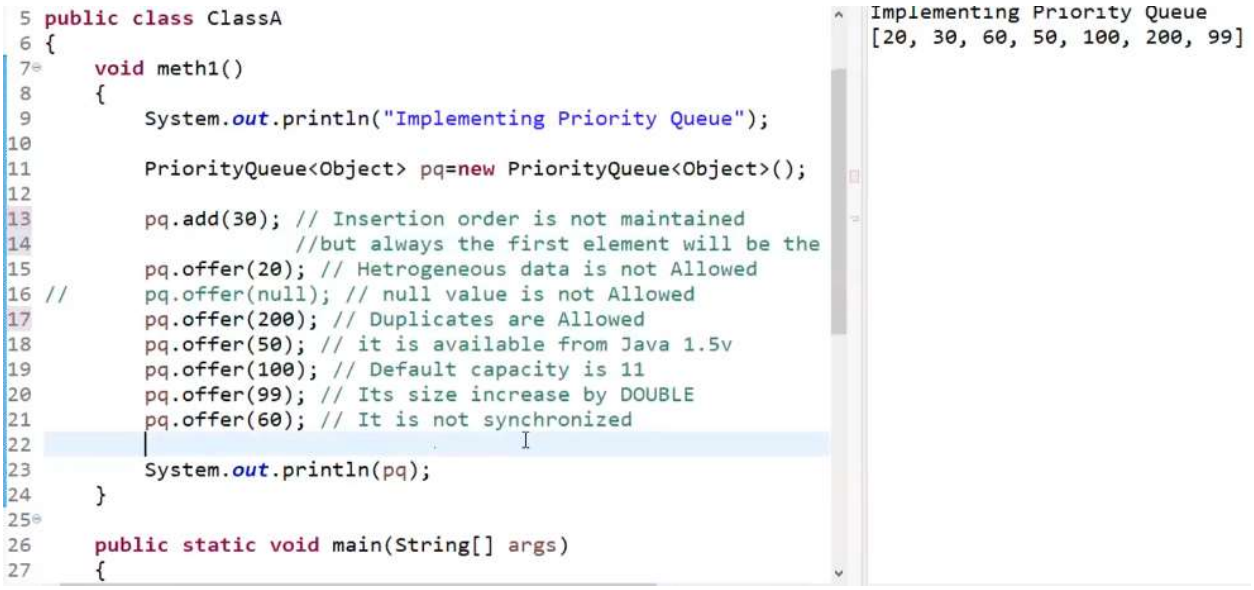
Why is it because heterogenous data is not allowed.



```
c class ClassA

oid meth1()

    System.out.println("Implementing Priority Queue");

    PriorityQueue<Object> pq=new PriorityQueue<Object>();

    pq.add(10); // Insertion order is
    pq.offer(20); // Hetrogeneous data is not Allowed
    pq.offer(null); // null value is
    pq.offer(10); // Duplicates are
    pq.offer(50); // it is available from Java 1.5v
    pq.offer(100); // Default capacity is 11
    pq.offer(99); // Its size increase by DOUBLE
    pq.offer(1); // It is not synchronized

    System.out.println(pq);


ublic static void main(String[] args)

     ClassA aobj=new ClassA();
```

```
Implementing Priority Queue
Exception in thread "main" java.lang.NullPoint
        at java.base/java.util.PriorityQueue.o
        at Training/com.pack1.ClassA.meth1(Cla
        at Training/com.pack1.ClassA.main(Clas
```

In the priority queue the first element we get is the smallest element, the remaining element does not follow.



```
public class ClassA
{
    void meth1()
    {
        System.out.println("Implementing Priority Queue");

        PriorityQueue<Object> pq=new PriorityQueue<Object>();

        pq.add(30); // Insertion order is not maintained
                    //but always the first element will be the
        pq.offer(20); // Hetrogeneous data is not Allowed
//      pq.offer(null); // null value is not Allowed
        pq.offer(200); // Duplicates are Allowed
        pq.offer(50); // it is available from Java 1.5v
        pq.offer(100); // Default capacity is 11
        pq.offer(99); // Its size increase by DOUBLE
        pq.offer(60); // It is not synchronized

        System.out.println(pq);
    }

    public static void main(String[] args)
    {
```

```
Implementing Priority Queue
[20, 30, 60, 50, 100, 200, 99]
```

```java
import java.util.PriorityQueue;

public class ClassA
{
    void meth1()
    {
        System.out.println("Implementing Priority Queue");

        PriorityQueue<Object> pq=new PriorityQueue<Object>();

        pq.add(30); // Insertion order is not maintained
                    //but always the first element will be the smallest element
        pq.offer(27); // Hetrogeneous data is not Allowed
//      pq.offer(null); // null value is not Allowed
        pq.offer(20); // Duplicates are Allowed
        pq.offer(5); // it is available from Java 1.5v
        pq.offer(1); // Default capacity is 11
        pq.offer(3); // Its size increase by DOUBLE
        pq.offer(60); // It is not synchronized

        System.out.println(pq);

        System.out.println("remove(): "+ pq.remove());
        System.out.println(pq);

        System.out.println("poll(): "+pq.poll());
//      pq.clear();
//      System.out.println("remove(): "+ pq.remove()); //It generates NoSuchElementException
//      System.out.println("poll(): "+pq.poll());

        System.out.println("peek(): "+ pq.peek());
        System.out.println(pq);
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();
    }
}
```