

# Java Queue Interface

Java Queue interface orders the element in FIFO(First In First Out) manner. In FIFO, first element is removed first and last element is removed at last.

## Queue Interface declaration

```
public interface Queue<E> extends Collection<E>
```

## Methods of Java Queue Interface

Method	Description
boolean add(object)	It is used to insert the specified element into this queue and return true upon success.
boolean offer(object)	It is used to insert the specified element into this queue.
Object remove()	It is used to retrieves and removes the head of this queue.
Object poll()	It is used to retrieves and removes the head of this queue, or returns null if this queue is empty.
Object element()	It is used to retrieves, but does not remove, the head of this queue.
Object peek()	It is used to retrieves, but does not remove, the head of this queue, or returns null if this queue is empty.

## PriorityQueue class

The PriorityQueue class provides the facility of using queue. But it does not orders the elements in FIFO manner. It inherits AbstractQueue class.

## PriorityQueue class declaration

Let's see the declaration for java.util.PriorityQueue class.

```
public class PriorityQueue<E> extends AbstractQueue<E> implements Serializable
```

## Java PriorityQueue Example

```
import java.util.*;
class TestCollection12{
public static void main(String args[]){
    PriorityQueue<String> queue=new PriorityQueue<String>();
    queue.add("Amit");
    queue.add("Vijay");
    queue.add("Karan");
    queue.add("Jai");
    queue.add("Rahul");
    System.out.println("head:"+queue.element());
    System.out.println("head:"+queue.peek());
    System.out.println("iterating the queue elements:");
    Iterator itr=queue.iterator();
    while(itr.hasNext()){
        System.out.println(itr.next());
    }
    queue.remove();
    queue.poll();
    System.out.println("after removing two elements:");
    Iterator<String> itr2=queue.iterator();
    while(itr2.hasNext()){
        System.out.println(itr2.next());
    }
}
}
```

### Test it Now

```
Output:head:Amit
        head:Amit
        iterating the queue elements:
```

```
Amit
Jai
Karan
Vijay
Rahul
after removing two elements:
Karan
Rahul
Vijay
```

## Java PriorityQueue Example: Book

Let's see a PriorityQueue example where we are adding books to queue and printing all the books. The elements in PriorityQueue must be of Comparable type. String and Wrapper classes are Comparable by default. To add user-defined objects in PriorityQueue, you need to implement Comparable interface.

```
import java.util.*;

class Book implements Comparable<Book>{
    int id;
    String name,author,publisher;
    int quantity;
    public Book(int id, String name, String author, String publisher, int quantity) {
        this.id = id;
        this.name = name;
        this.author = author;
        this.publisher = publisher;
        this.quantity = quantity;
    }
    public int compareTo(Book b) {
        if(id>b.id){
            return 1;
        }else if(id<b.id){
            return -1;
        }else{
            return 0;
        }
    }
}
```

```
}  
  
public class LinkedListExample {  
  
public static void main(String[] args) {  
    Queue<Book> queue=new PriorityQueue<Book>();  
    //Creating Books  
    Book b1=new Book(121,"Let us C","Yashwant Kanetkar","BPB",8);  
    Book b2=new Book(233,"Operating System","Galvin","Wiley",6);  
    Book b3=new Book(101,"Data Communications & Networking","Forouzan","Mc Graw Hill",4);  
    //Adding Books to the queue  
    queue.add(b1);  
    queue.add(b2);  
    queue.add(b3);  
    System.out.println("Traversing the queue elements:");  
    //Traversing queue elements  
    for(Book b:queue){  
        System.out.println(b.id+" "+b.name+" "+b.author+" "+b.publisher+" "+b.quantity);  
    }  
    queue.remove();  
    System.out.println("After removing one book record:");  
    for(Book b:queue){  
        System.out.println(b.id+" "+b.name+" "+b.author+" "+b.publisher+" "+b.quantity);  
    }  
}  
}
```

Output:

```
Traversing the queue elements:  
101 Data Communications & Networking Forouzan Mc Graw Hill 4  
233 Operating System Galvin Wiley 6  
121 Let us C Yashwant Kanetkar BPB 8  
After removing one book record:  
121 Let us C Yashwant Kanetkar BPB 8  
233 Operating System Galvin Wiley 6
```

## Share this page



## Latest 4 Tutorials

