**Write a program in Java to insert and remove elements in a queue**

**package** queue;

**class** Node {

**int** data;

Node next;

Node(**int** data) {

**this**.data = data;

**this**.next = **null**;

}

}

**public** **class** QueueOperations {

Node front,rear;

**public** QueueOperations() {

**this**.front = **null**;

**this**.rear = **null**;

}

**public** **void** enqueue(**int** data) {

Node newNode = **new** Node(data);

**if**(**this**.rear == **null**) {

**this**.front = **this**.rear = newNode;

}

**else** {

**this**.rear.next = newNode;

**this**.rear = newNode;

}

}

**public** **void** dequeue() {

**if**(**this**.front == **null**) {

System.***out***.println("Queue is empty!");

}

**else** {

Node temp = **this**.front;

**this**.front = **this**.front.next;

**if**(**this**.front == **null**) {

**this**.rear = **null**;

}

}

}

**public** **void** print() {

**if**(**this**.front == **null**) {

System.***out***.println("Queue is empty!");

}

**else** {

System.***out***.println("Queue: ");

Node node = **this**.front;

**while**(node != **null**) {

System.***out***.print(node.data);

node = node.next;

**if**(node != **null**) {

System.***out***.print("->");

}

}

System.***out***.println();

}

}

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

QueueOperations qo = **new** QueueOperations();

qo.enqueue(2);

qo.enqueue(4);

qo.enqueue(6);

qo.enqueue(8);

qo.enqueue(10);

qo.print();

qo.dequeue();

qo.print();

qo.dequeue();

qo.print();

}

}