Nitin Sikarwar
Sec - C(38) [2015004511]

DATE:

public class stack { STACK using Linkedlist

```
import java.util.*;
public class stack{
    static class Node
    static Node head;
    static class Node{
        int data;
        Node next;
        Node (int data){
            this.data = data;
            this.next = next;
        }
    }
public void push (int data){
    Node n = new Node (data);
    if(isEmpty())
    {
        head = n;
        return;
    }
    n.next = head;
    head = n;
}
public boolean isEmpty(){
    int r=0;
    if(head == null)
        return true;
    return false;
}
```

Nitin Sikarwal
Sec-C(38) [2015004517]

Dhoom
PAGE NO.:
DATE: / /

```
public int peek() {
    if(isEmpty()) {
        System.out.println("Invalid operation");
        return ;
    }
    int l = head.data;
    return l;
}
public int pop() {
    if(isEmpty()) {
        S.o.pl("Invalid operation");
        return -1;
    }
    else {
        int l = head.data;
        head = head.next;
        return l;
    }
}
public void display() {
    Node curr = head;
    if(curr =
    while (curr != null) {
        S.o.p(curr.data);
        curr = curr.next;
    }
}
public static void main(String args[]) {
    Stack S = new Stack();
    S.push(2);
    S.push(3);
    S.pop();
    S.display;
}
```

Nitin Sikarwar
Sec - C (30)

PAGE NO.:
DATE:

// Stack using Array :-

```
import java.util.*;
Public class Stack {
    static final int max = 5;
    int arr[] = new int [max];
    int top;
    Stack () {
        top = -1;
    }
    public boolean isEmpty() {
        if (top == -1) {
            return true;
        }
        else return false;
    }
    public boolean isFull() {
        if (top >= (max-1)) {
            return true;
        }
        return false;
    }
    public void push (int x) {
        if isFull() {
            S.o.p("Stack Overflow");
        }
        else
        {
            top ++;
            arr [top] = x;
        }
    }
```

Nitin Sikarwar
Sec- C(38)

Dhoom
PAGE NO.:
DATE: / /

```
public int pop() {
    if(isEmpty()) {
        S.o.p("Stack Underflow");
        return -1;
    }
    else {
        int d= arr[top];
        top--;
        return d;
    }
}
public int peek() {
    if(isEmpty()) {
        S.o.p("Stack underflow");
        return -1;
    }
    else {
        int d= arr[top];
        return d;
    }
}
public void display() {
    for(int i=top; i>=0; i--) {
        S.o.p(arr[i]);
    }
}
public static void main(String args[]) {
    Stack s= new Stack();
    s.push(1);
    s.push(2);
    s.print();
}
}
```

Nitin Sikarwar
Sec-C (38)

PAGE NO.:
DATE:

Queue using LinkedList :—

```
import java.util.*;
public class Queue {
    class Node {
        int data;
        Node next;
        Node (int data) {
            this.data = data;
            this.next = next;
        }
    static Node front;
    static Node rear;
    public boolean isEmpty() {
        if front == null) {
            return true; }
        return false;
    }
    public void enqueue (int data) {
        Node n = new Node (data);
        if (isEmpty()) {
            front = n;
            rear = n;
            return;
        }
        rear.next = n;
        rear = n;
    }
```

Nitin Sikarwar
Sec-C (38)

Dhoom
PAGE NO.:
DATE: / /

```
public int dequeue(){
        if(isEmpty()){
            S.o.p("Queue is empty");
            return -1;
        }
        int x = front.data;
        front = front.next;
        return x;
}
public int front()
{
        if(isEmpty()){
            S.o.p("Queue is Empty");
            return -1;
        }
        return front.data;
}
public int rear(){
        if(isEmpty()){
            S.o.p("Queue is Empty");
            return -1;
        }
        return rear.data;
}
public void display(){
        Node curr = front;
        while(curr!=null){
            S.o.p(curr.data);
            curr = curr.next;
        }
}
```

Nitin Sikarwar
Sec-C(38)

```
public static void main (String args[]){
        Queue q = new Queue();
        q.enqueue(1);
        q.enqueue(2);
        q.enqueue(3);
        q.enqueue(4);
        q.dequeue();
        q.display();
    }
}
```

Queue using Array :-

```
import java.util.*;
public class QueueAR {
    static final int max=5;
    int arr[] = new int [max];
    static int front;
    static int rear;
    QueueAR() {
        front = 0;
        rear = 0;
    }
    public boolean isEmpty(){
        if (rear == front) {
            return true;
        }
        else {
            return false;
        }
}
```

© Nitin Sikarwar
Sec- C (30)

Dhoom
PAGE NO.:
DATE: / /

```
public boolean isFull(){
    if(rear-1) == (max-1))
    {
        return true;
    }
    return false;
}
public void enqueue(int x){
    if(isFull()){
        s.o.p("Queue is Full");
    }
    else{
        arr[rear] = x;
        rear++;
    }
}
public void dequeue(){
    if(isEmpty()){
        s.o.p("Queue is Empty");
    }
    else{
        int d= arr[front];
        front++;
    }
}
public int front(){
    if(isEmpty()){
        s.o.p("Queue is Empty");
        return -1;
    }
```

Nitin Sikarwar
Sec- C(38)

```
        else {
            return arr[front];
        }
    }
    public int rear() {
        if(isEmpty()) {
            S.o.p("Queue is Empty");
            return -1;
        }
        else {
            return arr[rear-1];
        }
    }
    public void display() {
        for(int i=front; i<=rear-1; i++) {
            S.o.pl arr[i];
        }
    }
    public static void main(String args[]) {
        Queue8 Q = new Queue8();
        Q.enqueue(1);
        Q.enqueue(2);
        Q.enqueue(3);
        Q.dequeue();
        Q.display();
    }
}
```

Thank You