

Q Implement Interface Queue Operations

```
import java.util.*;  
interface Q
```

```
{  
    void insert_rear(int item);  
    int delete_front();  
    void display();  
}
```

```
class Queue implements Q
```

```
{  
    private int q[];  
    private int rear;  
    private int front;  
    Queue(int size)
```

```
{  
        q = new int[size];  
        rear = -1;  
        front = 0;  
    }
```

```
    public void insert_rear(int item)
```

```
{  
    if (rear == q.length - 1)  
        System.out.println("Queue Overflow");  
    else
```

```
        q[++rear] = item;
```

```
    public int delete_front()
```

```
{  
    if (front > rear)
```

NAME - MUSKAN GUPTA

USN - 1BM19C5091

Date / /



```
System.out.println("Queue Underflow");  
front front = 0;  
front rear = item & rear = -1;  
return -1;  
}  
return q[front++];  
}
```

```
public void display()  
{  
    System.out.println("Contents of queue:");  
    for (int i = front; i <= rear; i++)  
        System.out.println(q[i] + " ");  
    System.out.println();  
}
```

```
class JMain  
{
```

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

```
        Queue obj = new Queue (10);
```

```
        int n, item;
```

```
        Scanner sc = new Scanner (System.in);
```

```
        while (true)
```

```
        {
```

```
            System.out.println("1. Insert into queue\n2. Delete from queue\n3. Display\n4. Exit");
```

```
            n = sc.nextInt();
```

```
            switch (n)
```

```
            {  
                Case 1: System.out.println("Enter item");
```



NAME - MUSKAN GUPTA

USN - IBM19C5091



```
item = sc.nextInt();  
obj.insert_rear(item);  
break;
```

```
Case 2: item = obj.delete_front()
```

```
if (item == -1)  
System.out.println("queue is empty");  
else
```

```
System.out.println("deleted item: " + item);  
break;
```

```
Case 3 : obj.display();  
break;
```

```
default : System.exit(0);  
}
```

```
}
```

```
}
```

```
}
```



2. Write a Java Program to compute the factorial of a number. The input value must be tested for validity. If it is greater than 15, the method `computeFactorial()` should raise an userdefined Exception `MyException` with appropriate messages

```
import java.util.*;
```

```
class MyException extends Exception
```

```
{
    int num;
    MyException (int n)
    {
```

```
        num = n;
    }
```

```
    public String toString ()
    {
```

```
        return "The input number cannot be greater than 15";
    }
```

```
}
```

```
public class Factorial - Excep
```

```
{
    int computeFact (int n) throws MyException
    {
```

```
        if ( n > 15)
        {
```

```
            throw new MyException (n);
        }
```

```
        else if ( n == 0)
        {
```



NAME - MUSKAN GUPTA

USN - 18BM19CS091



```
        return 1;
    }
    else
    {
        return n * Computefact (n-1);
    }
}

public static void main (String args[])
{
    Scanner sc = new Scanner (System.in);
    int n, fact;
    for (int i = 0; i < 2; i++)
    {
        System.out.println ("Enter the number:");
        n = sc.nextInt();
        Factorial - Excep f = new Factorial - Excep
        try
        {
            fact = f.Computefact(n);
            System.out.println ("The factorial of "
                                + n + " is " + fact);
        }
        catch (MyException e)
        {
            System.out.println ("Caught Exception: " + e);
        }
    }
}
```





Q Write a java program to create an account class. Define appropriate constructor for this class. Implement a separate methods to display account balance and withdraw money. Raise a user defined exception if there is an attempt to withdraw money which is greater than the account balance. Make necessary assumptions required. \*

```
import java.util.*;
class MyException extends Exception
{
    double amount;
    MyException(double a)
    {
        amount = a;
    }
    public String toString()
    {
        return "Insufficient balance in your account\n Your account balance = "+amount;
    }
}
class Account
{
    Scanner sc = new Scanner(System.in);
    double balance;
    int amt;
    Account(double bal)
    {
        balance = bal;
    }
}
```



NAME - MUSKAN GUPTA

USN - IBM, 19, C5091



NAME  
USN

double withdraw () throws MyException

{  
    System.out.println("Enter the amount to withdraw");

    amt = sc.nextInt();  
    if (balance >= amt)

    {  
        balance = balance - amt;  
        return balance;

    }  
    else  
    {  
        throw new MyException(balance);

void display ()

{  
    System.out.println("Account Balance = " + balance);  
}

class AccMain

{  
    public static void main (String args [])

    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the initial balance");  
        double b = sc.nextDouble();

        Account obj = new Account(b);  
        while (true)

        {  
            System.out.println("1. Withdraw\n2. Display Balance\n3. Exit")

NAME - MUSKAN GUPTA

USN - 18M19C3091

Date \_\_/\_\_/\_\_



```
System.out.println("Enter the choice");  
int n = sc.nextInt();
```

```
switch (n)
```

```
{
```

```
    case 1:
```

```
        try
```

```
        {
```

```
            obj.withdraw();
```

```
        }
```

```
        catch (MyException e)
```

```
        {
```

```
            System.out.println(e);
```

```
        }
```

```
        break;
```

```
    case 2:
```

```
        obj.display();
```

```
        break;
```

```
    default:
```

```
        System.exit(0);
```

```
    }
```

```
}
```

```
}
```

```
}
```