Week 10 – Extra Programs

1. Implement Interfaces – QUEUE OPERATIONS

```
import java.util.*;
interface IntQueue {
    void insert_rear(int item);

    int delete_front();
    void displayQ();
    }



 class Queue implements IntQueue {
 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 a) {
 if(rear==q.length-1)
 System.out.println("Queue is full.");
 else
 q[++rear] = a;
 }


 public int delete_front() {
 if(front>rear)
    {
        front=0;
        rear=-1;
        return -1;
    }
    return q[front++];
 }
public void displayQ()
{
    int i;
    if(front>rear)
    {
        System.out.println("Queue is empty\n");
        return ;
    }
    System.out.println("Contents of queue\n");
    for(i=front;i<=rear;i++)
```

```
    {
        System.out.println(q[i]);
    }
}
    }


class QueueInter {
 public static void main(String args[]) {
Scanner ss=new Scanner(System.in);
Queue myqueue = new Queue(3);
int choice;
    for(;;)
    {
        System.out.println("\n1:Insert rear\n2:Delete
front\n3:Display\n4:exit\n");
        System.out.println("Enter the choice");
        choice=ss.nextInt();
        switch(choice)
        {
            case 1:System.out.println("Enter the item to be inserted");
            int item=ss.nextInt();
            myqueue.insert_rear(item);
            break;
            case 2:item=myqueue.delete_front();
            if(item==-1)
                System.out.println("Queue is empty\n");
            else
```

```
                System.out.println("Item deleted="+item);
            break;
            case 3:myqueue.displayQ();
            break;
            default:System.exit(0);
        }
    }


 }
}
```

OUTPUT:

```
C:\Users\win10\Documents\Java lab programs>javac QueueInter.java

C:\Users\win10\Documents\Java lab programs>java QueueInter

1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
1
Enter the item to be inserted
10

1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
1
Enter the item to be inserted
20

1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
1
Enter the item to be inserted
30

1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
1
Enter the item to be inserted
40
Queue is full.
```

```
1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
3
Contents of queue

10
20
30

1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
2
Item deleted=10

1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
3
Contents of queue

20
30

1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
2
Item deleted=20
```

```
1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
2
Item deleted=30

1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
2
Queue is empty


1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
3
Queue is empty


1:Insert rear
2:Delete front
3:Display
4:exit

Enter the choice
4
```

2. Write a Java program to compute the factorial of a number. The input value must be testedfor validity. If it is greater than 15, the method ComputeFactorial( ) should raise an Userdefined Exception MyException with appropriate messages.

```
import java.util.Scanner;

class MyException extends Exception {
    int num;

    MyException(int x) {
        num = x;
    }

    public String toString() {
        return "Number Entered " + num + " is invalid.";
    }
}

class Factorial {
    static int ComputeFactorial(int n) throws MyException {
        if(n > 15){
            throw new MyException(n);
        }
        else{
        if (n == 0)
            return 1;
        else
```

```
            return (n * ComputeFactorial(n - 1));
        }
    }

    public static void main(String args[]) {

        Scanner s = new Scanner(System.in);
        int i, fact = 1;
        System.out.println("Enter a number under 15:");
        int number = s.nextInt();
        try {
            fact = ComputeFactorial(number);
            System.out.println("Factorial of " + number + " is: " + fact);

        }
        catch (MyException e) {
            System.out.println(e);
        }

    }
}
```

OUTPUT:

```
C:\Users\win10\Documents\Java lab programs>java Factorial
Enter a number under 15:
5
Factorial of 5 is: 120

C:\Users\win10\Documents\Java lab programs>javac Factorial.java

C:\Users\win10\Documents\Java lab programs>java Factorial
Enter a number under 15:
17
Number Entered 17 is invalid.
```

3. 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. than the account balance. Make necessary assumptions required.

```
import java.util.Scanner;


 class Insufficient extends Exception {


    double amount;
    Insufficient(double amount) {
      this.amount = amount;
   }


   public String toString() {
    return "INSUFFICIENT BALANCE\nYOUR ACCOUNT
BALANCE="+amount;
}
}
```

```
class ACCOUNT{
    Scanner s=new Scanner(System.in);
    double balance;
    int amt;
    long acc;
    ACCOUNT(double balance,long acc)
       {
         this.balance=balance;
         this.acc=acc;
       }
     double withdraw() throws Insufficient
     {
         System.out.println("ENTER THE AMOUNT TO BE
WITHDRAWN");
         amt=s.nextInt();
         if(balance>=amt)
         {
            balance=balance-amt;
            return balance;
         }
         else
         throw new Insufficient(balance);
     }

   void display(){
    System.out.println("ACCOUNT BALANCE="+balance);
}
    }
```

```
class accmain{
    public static void main(String args[])
    {
       Scanner s=new Scanner(System.in);
     System.out.println("ENTER THE INITIAL BALANCE");
     double b=s.nextDouble();
     System.out.println("ENTER THE ACCOUNT NUMBER");
     long l=s.nextLong();
     ACCOUNT acc= new ACCOUNT(b,l);
              for(;;) {
     System.out.println("1-WITHDRAWAL\n2-DISPALY BALANCE\n3-
EXIT");
     System.out.println("ENTER THE CHOICE");
     int c=s.nextInt();
     switch(c)
     {
        case 1:
        try{
           acc.withdraw();
        }catch(Insufficient e)
        {
           System.out.println(e);
        }
        break;
        case 2:
        acc.display();
                    break;
```

```
        case 3:
        System.exit(0);
        default:
        System.out.println("INVALID CHOICE");
     }
    }
  }
}
```

OUTPUT:

```
C:\Users\win10\Documents\Java lab programs>javac accmain.java

C:\Users\win10\Documents\Java lab programs>java accmain
ENTER THE INITIAL BALANCE
10000
ENTER THE ACCOUNT NUMBER
123456
1-WITHDRAWAL
2-DISPALY BALANCE
3-EXIT
ENTER THE CHOICE
1
ENTER THE AMOUNT TO BE WITHDRAWN
5000
1-WITHDRAWAL
2-DISPALY BALANCE
3-EXIT
ENTER THE CHOICE
2
ACCOUNT BALANCE=5000.0
1-WITHDRAWAL
2-DISPALY BALANCE
3-EXIT
ENTER THE CHOICE
1
ENTER THE AMOUNT TO BE WITHDRAWN
500
1-WITHDRAWAL
2-DISPALY BALANCE
3-EXIT
ENTER THE CHOICE
2
ACCOUNT BALANCE=4500.0
1-WITHDRAWAL
2-DISPALY BALANCE
3-EXIT
ENTER THE CHOICE
3
```

```
C:\Users\win10\Documents\Java lab programs>javac accmain.java

C:\Users\win10\Documents\Java lab programs>java accmain
ENTER THE INITIAL BALANCE
5000
ENTER THE ACCOUNT NUMBER
789456
1-WITHDRAWAL
2-DISPALY BALANCE
3-EXIT
ENTER THE CHOICE
1
ENTER THE AMOUNT TO BE WITHDRAWN
6000
INSUFFICIENT BALANCE
YOUR ACCOUNT BALANCE=5000.0
1-WITHDRAWAL
2-DISPALY BALANCE
3-EXIT
ENTER THE CHOICE
2
ACCOUNT BALANCE=5000.0
1-WITHDRAWAL
2-DISPALY BALANCE
3-EXIT
ENTER THE CHOICE
3
```