School of Information Technology & Engineering

Winter Semester 2021-2022

ITA5004 – Object Oriented Programming using JAVA

DIGITAL ASSIGNMENT THEORY -02

NAME:PRIYADHARSHINI.R

REGNO:21MCA0032

SLOT:A2

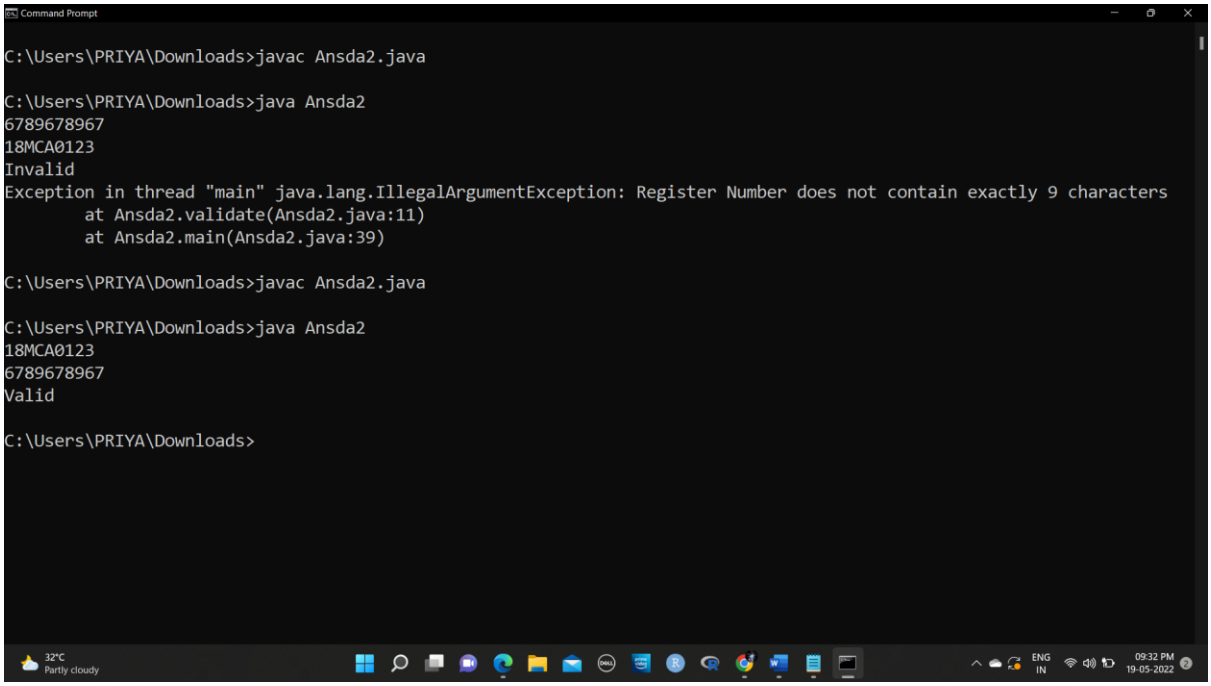
1. Write a Java program to read the Register Number and Mobile Number of a student. Create user-defined exception and handle the following: 1. If the Register Number does not contain exactly 9 characters in specified format(2 numbers followed by 3 characters followed by 4 numbers) or if the Mobile Number does not contain exactly 10 characters, throw an IllegalArgumentException. 2. If the Mobile Number contains any character other than a digit, raise a NumberFormatException. 3. If the Register Number contains any character other than digits and alphabets, throw a NoSuchElementException. 4. If they are valid, print the message 'valid' else 'Invalid'.

CODE:

```
import java.util.NoSuchElementException;

import java.util.Scanner;

import java.util.regex.Matcher;

import java.util.regex.Pattern;


public class Ansda2{


   static void validate(String r, String n){

      if(r.length() != 9){

         System.out.println("Invalid");

         throw new IllegalArgumentException("Register Number does not contain exactly 9 characters");

      }

      if(n.length() != 10){

         System.out.println("Invalid");
```

```
        throw new IllegalArgumentException("Mobile Number does not contain exactly 10 characters");
      }

      String pattern = "^[6|7|8|9]{1}\\d{9}";
      Pattern a = Pattern.compile(pattern);
      Matcher m1 = a.matcher(n);
      if(!m1.find()){
        throw new NumberFormatException("Mobile Number cannot contain any character other than a digit");
      }

      String pattern2 = "^[1-9]{2}[A-Z]{3}[0-9]{4}$";
      Pattern b = Pattern.compile(pattern2);
      Matcher m2 = b.matcher(r);
      if(!m2.find()){
        throw new NoSuchElementException("Registration Number cannot contain any character other than digits and alphabets");
      }

   }

   public static void main(String args[]){
      Scanner sc = new Scanner(System.in);
      String reg = sc.nextLine();
      String no = sc.nextLine();
      sc.close();
      validate(reg, no);
      System.out.println("Valid");
   }
}
```

OUTPUT:



2. Write a simulation program for the fruit market. The farmer will be able to produce different types of fruits (apple, orange, grape, and watermelon), and put them in the market to sell. The market has limited capacity and farmers have to stand in a queue if the capacity is exceeded to sell their fruits. Consumers can come to the market any time and purchase their desired fruits; and if the fruits they want to buy runs out, they are willing to wait until the supply of that kind is ready. (Hint: implementing this market will encounter the producer and consumer problem, and it probably needs multiple buffers for different kinds of fruits). Create your own necessary methods, variables to show the producer-consumer problem

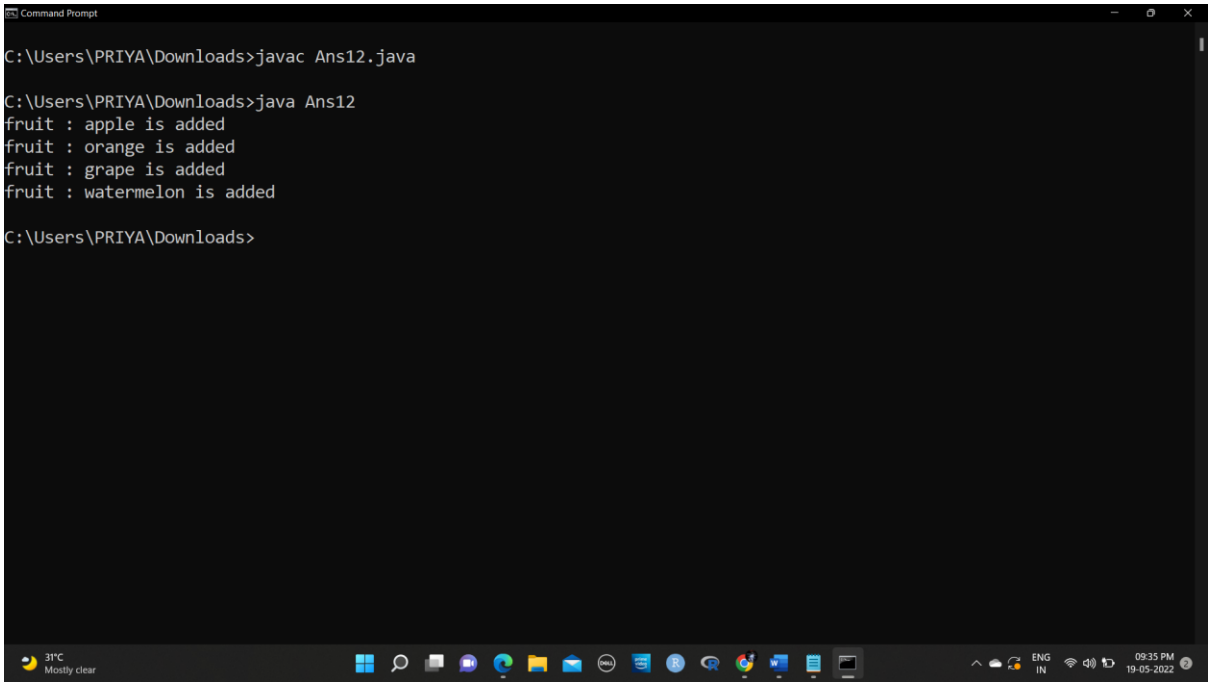
CODE:

```
import java.util.ArrayList;

public class Ans12

{

public static void main(String[] args)

{

                Market sm = new Market(25);

                sm.farmer("apple");

                sm.farmer("orange");

                sm.farmer("grape");

                sm.farmer("watermelon");

                sm.consumer();

}
```

```
}
class Market{
private ArrayList<String> fts = new ArrayList<>();
private int num;
public Market(int num)
{
if (num > 0)
{
this.num = num;
}
else
{
throw new IllegalArgumentException("Invalid argument");
}
}
private synchronized boolean isFull()
{
return fts.size() == this.num;
}
private synchronized boolean isEmpty()
{
return fts.isEmpty();
}
public synchronized void farmer(String ft)
{
if (isFull())
{
System.out.println("No more fruits can be accepted");
try
{
wait();
```

```
}
catch(InterruptedException e)
{
System.out.println("Interruption");
}
}
fts.add(ft);
System.out.printf("fruit : %s is added %n", ft);
}
public synchronized String consumer()
{
if (isEmpty())
{
System.out.println("No good fruit");
try
{
wait();
}
catch (InterruptedException e)
{
System.out.println("Interruption");
}
}
String cur_fts = fts.remove(0);
notifyAll();
return cur_fts;
}
}
```

OUTPUT:


Command Prompt
C:\Users\PRIYA\Downloads>javac Ans12.java
C:\Users\PRIYA\Downloads>java Ans12
fruit : apple is added
fruit : orange is added
fruit : grape is added
fruit : watermelon is added
C:\Users\PRIYA\Downloads>
31°C
Mostly clear
ENG
IN
09:35 PM
19-05-2022