CSE1007 – Java Programming

LAB Assignment 3

Submitted by: - Shivam Tayal (18BIT0271)

EXCEPTION HANDLING

1. Read the Register Number and Mobile Number of a student. If the Register Number does not contain exactly 9 characters or if the Mobile Number does not contain exactly 10 characters, throw an IllegalArgumentException. If the Mobile Number contains any character other than a digit, raise a NumberFormatException. If the Register Number contains any character other than digits and alphabets, throw a NoSuchElementException. If they are valid, print the message 'valid' else 'invalid'

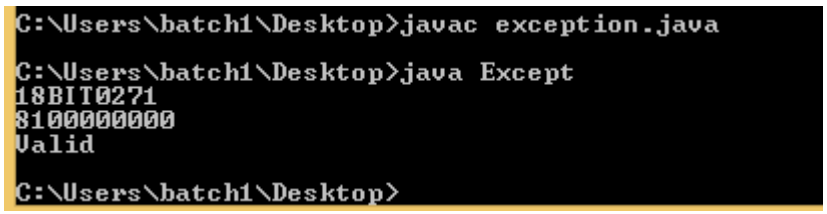
Code: -

```
import java.util.*;
import java.util.regex.*;
class Except
{
	public static void main(String args[])
	{
		boolean flag = true;
		String RegNo;
```

```
        String MobileNo;
        Scanner cin = new Scanner(System.in);
        try{
                RegNo = cin.nextLine();
                if(RegNo.length() != 9)
                        throw new IllegalArgumentException();
                else if(Pattern.matches("^\\w{9}$", RegNo)==false)
                        throw new NoSuchElementException();
                MobileNo = cin.nextLine();
                if(MobileNo.length()!=10)
                        throw new IllegalArgumentException();
                else if(Pattern.matches("^[0-9]{10}$", MobileNo)==false)
                        throw new NumberFormatException();
        }
        catch(Exception e)
        {
                System.out.println(e);
                System.out.println("Invalid");
                flag=false;
        }
        if(flag==true)
                System.out.println("Valid");
}
```

}

Output: -



2. A bag contains balls of 4 different colors- red, green, blue and yellow. Simulate picking up a ball at random for ten times. If the same colored ball is picked more than thrice, throw SameColorBallException and proceed with the simulation once again. After 10 valid picks, print the number of balls chosen from each of these colors.

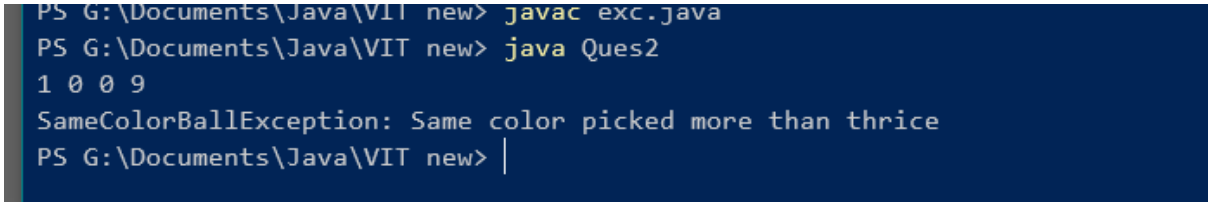
Code: -

```
import java.io.*;
import java.util.*;
class Ques2
{
	public static void main(String args[])
	{
		try
		{
			BallRandom br = new BallRandom();
```

```
                br.dox();
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
}
class BallRandom
{
    void dox() throws SameColorBallException
    {
        Random r = new Random();
        int count_r=0, count_g=0, count_b=0, count_y=0;
        for(int i=0; i<10;i++)
        {
            int ball = r.nextInt(4);
            if(ball==0)
                count_r++;
            else if(ball==0)
                count_g++;
            else if(ball==0)
                count_b++;
```

```
                else
                        count_y++;


        }
        System.out.println(count_r+" "+count_g+" "+count_b+" "+count_y);
        if(count_r>3 || count_b>3 || count_y>3 || count_g>3)
                throw new SameColorBallException("Same color picked more than thrice");
    }
}
class SameColorBallException extends Exception
{
    SameColorBallException(String s)
    {
        super(s);
    }
}
```

Output: -



PROGRAMS ON PACKAGES

1. Create an interface with methods add () and sub () in a package called 'pack1'. Create another package 'pack2' with an interface with methods multiply () and divide (). Write a main class to perform arithmetic operations on integer numbers by implementing both the interfaces.

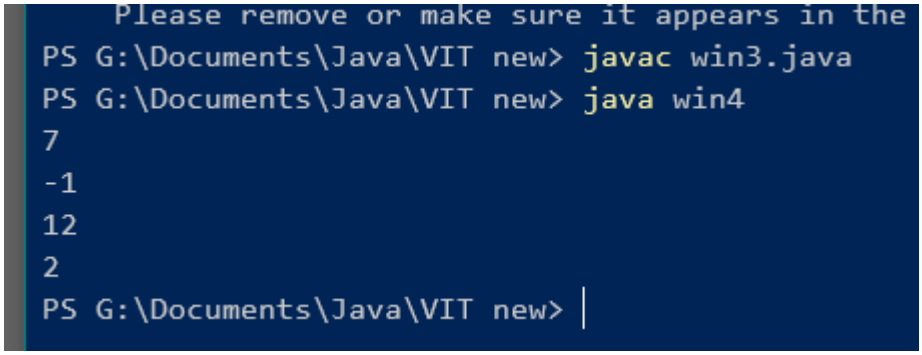
Code: -

```
package pack1;

public interface win

{

        public int add(int a, int b);

        public int sub(int a, int b);

}


package pack2;

public interface win2

{

        public int multiply(int a, int b);

        public int divide(int a, int b);

}



import pack1.*;

import pack2.*;
```

```
class win3 implements pack1.win,pack2.win2
{
	public int add(int a, int b)
	{
		return a+b;
	}
	public int sub(int a, int b)
	{
		return a-b;
	}
	public int multiply(int a, int b)
	{
		return a*b;
	}
	public int divide(int a, int b)
	{
		return a/b;
	}
}
class win4
{
	public static void main(String args[])
	{
```

```
            win3 w = new win3();
            System.out.println(w.add(3,4));
            System.out.println(w.sub(3,4));
            System.out.println(w.multiply(3,4));
            System.out.println(w.divide(4,2));
        }
}
```

Output: -



2. Write a program to demonstrate the knowledge of students in working with user-defined packages and sub-packages. Eg., Within the package named 'primespackage', define a class Primes which includes a method checkForPrime() for checking if the given number is prime or not. Define another class named TwinPrimes outside of this package which will display all the pairs of prime numbers whose difference is 2. (Eg, within the range 1 to 10, all possible twin prime numbers are (3,5), (5,7)). The TwinPrimes class should make use of the checkForPrime() method in the Primes class.

Code: -

```
package primespackage;

public class primes
```

```
{
	public static boolean checkForPrime(int a)
	{
		for(int i=2;i<a;i++)
		{
			if((a%i)==0)
			{
				return false;
			}
		}
		return true;
	}
}

import primespackage.*;
class TwinPrimes extends primespackage.primes
{
	public static void main(String args[])
	{
		int initial_range=2;
		int final_range=10;
		for(int i=initial_range;i<(final_range-2);i++)
		{
```

```
                    if(checkForPrime(i) && checkForPrime(i+2))
                    {
                        System.out.println(i+" "+(i+2));
                    }
                }
            }
}
```

Output: -

```
PS G:\Documents\Java\VIT new> javac -d . primes.java
PS G:\Documents\Java\VIT new> javac TwinPrimes.java
PS G:\Documents\Java\VIT new> java TwinPrimes
3 5
5 7
PS G:\Documents\Java\VIT new>
```

List of experiments in abstract class and interfaces

1. Write an abstract class special with an abstract method double Process (double P,double R). Create a subclass Discount and implement the Process() method with the following formula: net=PP*R/100. Return the Process() method with the following formula: total=P+P*R/100. Return the total.

Code: -

```
import java.util.*;

abstract class Net

{

    double p;

    double r;
```

```
        abstract void Process(double P,double R);
}

class Discount extends Net
{
        public void Process(double a,double b)
        {
                p=a;
                r=b;
        }
        public double getdata()
        {
                double net=p-p*r/100;
                return net;
        }
}

class Tax extends Discount
{
        public double getdata()
        {
                double total=p+p*r/100;
                return total;
```

```
    }
}


public class abstractdemo
{
    public static void main(String[] args)
    {
        Discount o1=new Discount();
        Tax o2=new Tax();
        o1.Process(5,6);
        o2.Process(5,6);
        System.out.println("Net="+o1.getdata());
        System.out.println("Total="+o2.getdata());
    }
}
```

Output: -

```
PS G:\Documents\Java\VIT new> javac abstractdemo.java
PS G:\Documents\Java\VIT new> java abstractdemo
Net=4.7
Total=5.3oid doxd() throws SameColorException
PS G:\Documents\Java\VIT new>
```

2. Write an interface called Numbers, with a method int Process(int x,int y). Write a class called Sum, in with the method Process finds the sum of two numbers and returns an int value. Write another class called Average, in which the process method finds the average of the two numbers and returns an int.

Code: -

```
interface Numbers
{
	public int process(int x,int y);
}

class Sum implements Numbers
{
	public int process(int x,int y)
	{
		return(x+y);
	}
}

class Average implements Numbers
{
	public int process(int x,int y)
	{
		return((x+y)/2);
	}
}

class InterfaceDemo
```

```
{
        public static void main(String args[])
        {
                int a,b;
                Sum add=new Sum();
                a=add.process(9,10);
                System.out.println("Your Sum is:"+a);
                Average avg=new Average();
                b=avg.process(90,90);
                System.out.println("Your Average is:"+b);
        }
}
```

Output: -

3. Write an interface called Exam with a method Pass ( ) that returns the total marks. Write another interface called Classify with a method Average (int total) which returns a string. Write a Class called Result which implements both Exam and Classify. The Pass method should get the marks from the user and finds the total marks and return. The Division method calculate the average marks and return “First” if the average is 60 or more, “SECOND” when average is 50 or more but below 60, “NO DIVISION” when average is less than 50.

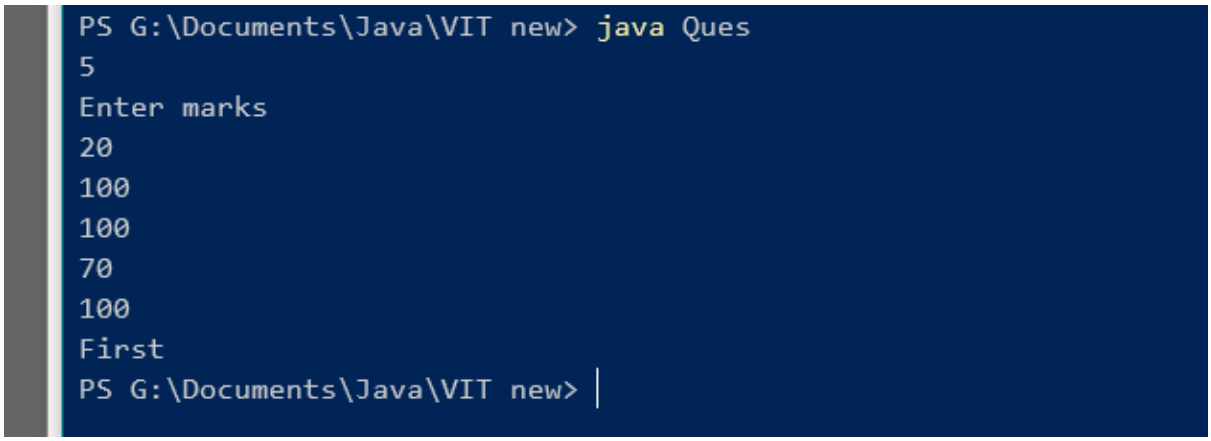
Code: -

```
import java.util.*;
interface Exam
{
	public int Pass();
}
interface Classify
{
	public String Average(int total);
}
class Result implements Exam,Classify
{
	int n=1;
	public int Pass()
	{
		Scanner cin = new Scanner(System.in);
		n = cin.nextInt();
		int total=0;
		System.out.println("Enter marks");
		for(int i=0;i<n;i++)
			total += cin.nextInt();
		return total;
	}
```

```
	public String Average(int tot)
	{
		int avg = tot/n;
		if(avg >60)
			return "First";
		else if(avg >50 && avg <=60)
			return "SECOND";
		else
			return "NO DIVISION";
	}
	public void Division(int tot)
	{
		System.out.println(Average(tot));
	}
}
class Ques
{
	public static void main(String args[])
	{
		Result res = new Result();
		int t = res.Pass();
		res.Division(t);
	}
```

}

Output: -


PS G:\Documents\Java\VIT new> java Ques
5
Enter marks
20
100
100
70
100
First
PS G:\Documents\Java\VIT new>