

Name: Abinash Satapathy

Reg. no.: 16BCE0081

Slot: L33 + L34

Subject: Java Programming (CSE1007) Lab

Experiment – 2

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.

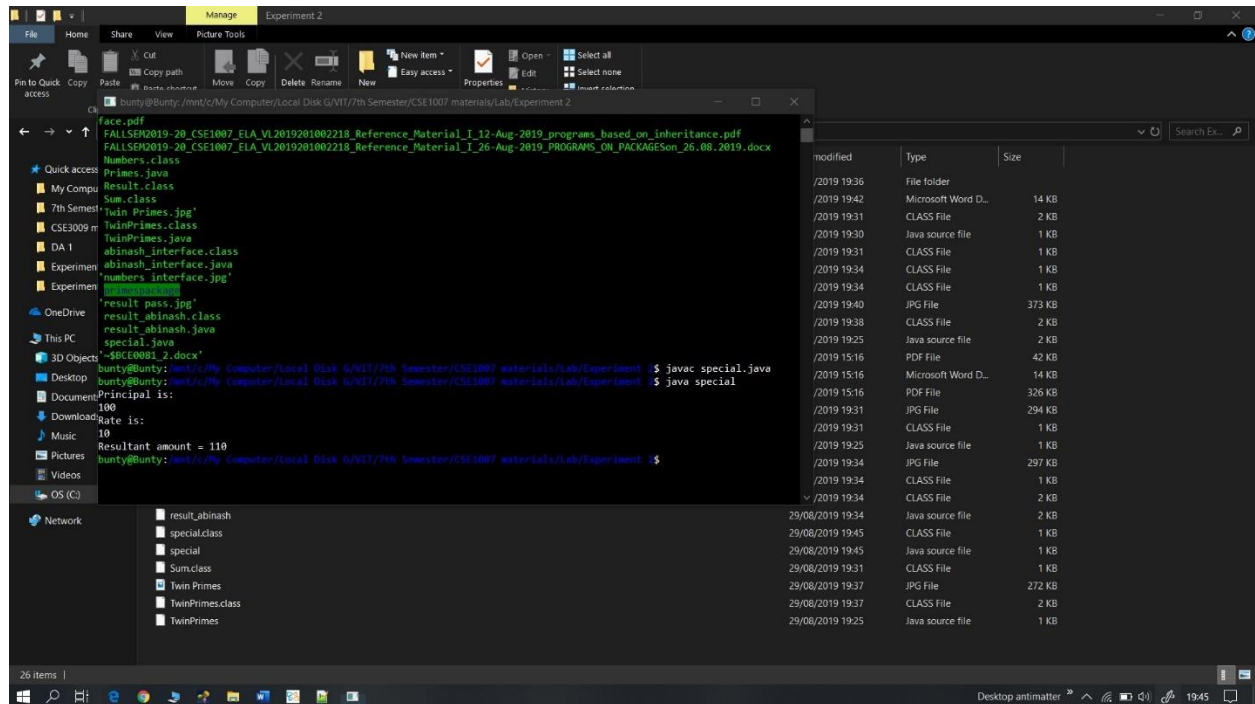
```
import java.util.*
abstract class special
{
    double process(double p, double r);
}

class Discount extends special
{
    double process(double p, double r)
    {
        return p+(p*r)/100;
    }
}

public class runmain
{
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);

        System.out.println("Enter Principal and Rate.");
        double principal = obj.nextDouble();
        double rate = obj.nextDouble();

        Discount d = new Discount();
        d.process(principal, rate);
    }
}
```



- Write an interface called Numbers, with a method `int Process(int x,int y)`. Write a class called Sum, in which 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.

interface Numbers

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

class Sum implements Numbers

```
{
    public int process(int x,int y)
    {
        return(x+y);
    }
}
```

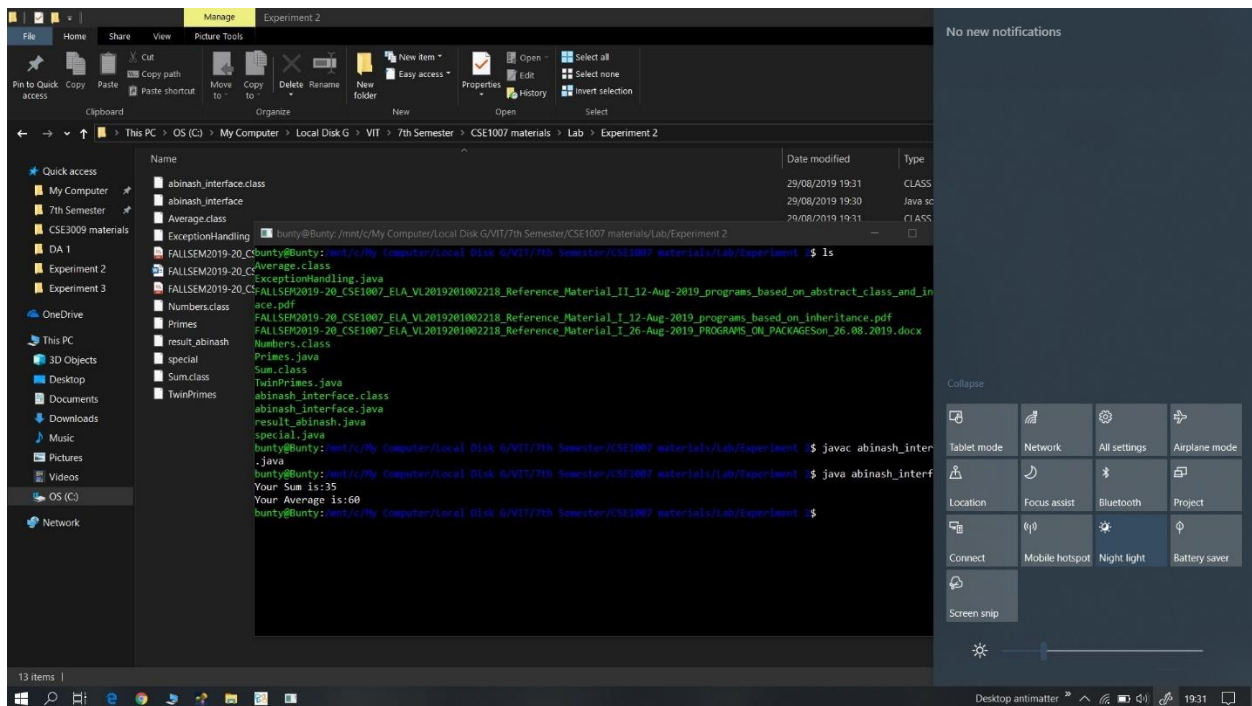
class Average implements Numbers

```
{
```

```

public int process(int x,int y)
{
return((x+y)/2);
}
}
class abinash_interface
{
public static void main(String args[])
{
int a,b;
Sum add=new Sum();
a=add.process(15,20);
System.out.println("Your Sum is:"+a);
Average avg=new Average();
b=avg.process(46,74);
System.out.println("Your Average is:"+b);
}
}

```



- 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.

```
import java.util.*;

interface Exam
{
    boolean Pass(int mark);
}

interface Classify
{
    String Division(int avg);
}

class Result implements Exam,Classify
{
    public boolean Pass(int mark)
    {
        if(mark>=50)
            return true;
        else
            return false;
    }

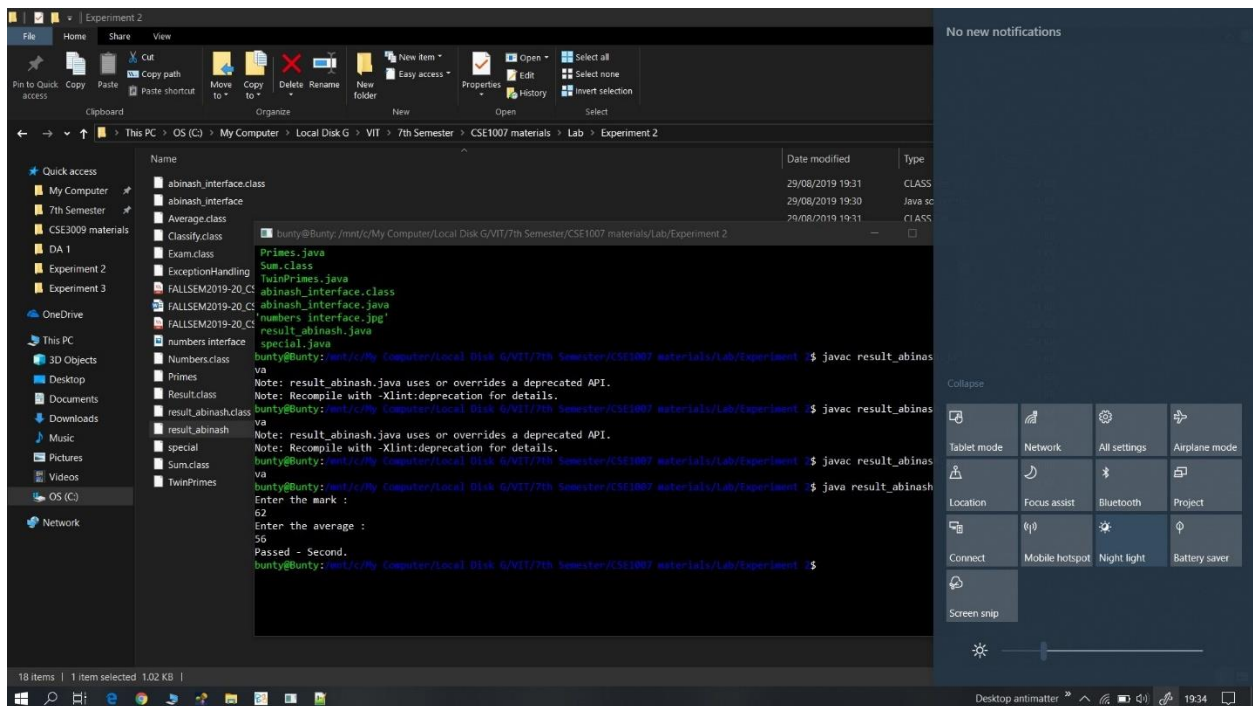
    public String Division(int avg)
    {
        if(avg>=60)
            return "First";
        else if(avg>=50)
            return "Second";
        else
            return "No-Division";
    }
}

public class result_abinash {
    public static void main(String[] args) {
        boolean pass;
        int mark,avg;
        String division;
```

```

Scanner in = new Scanner(System.in);
Result res=new Result();
try
{
    System.out.println("Enter the mark : ");
    mark=in.nextInt();
    System.out.println("Enter the average : ");
    avg=in.nextInt();
    pass=res.Pass(mark);
    division=res.Division(avg);
    if(pass)
        System.out.println("Passed - " + division + ".");
    else
        System.out.println("Failed - " + division+ ".");
}
catch(Exception e)
{
    System.out.println("Error : " + e);
}
}
}

```



4. 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.

Package:

```
package primespackage;
```

```
public class Primes{
```

```
    public static boolean checkForPrime(int a){
```

```
        if(a<4) return a>1;
```

```
        for(int i=2;i<a/2;i++) if (a%i==0) return false;
```

```
        return true;
```

```
    }
```

```
}
```

Code:

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

```
import java.io.*;
```

```
import primespackage.Primes;
```

```
class TwinPrimes{
```

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

    Scanner in=new Scanner(System.in);

    System.out.println("input limit to check for twin primes till:");

    int n=in.nextInt();

    if (n<5) return;

    for(int i=3;i+1<n;i+=4){

        if(Primes.checkForPrime(i)&&Primes.checkForPrime(i+2))
        System.out.println(i+", "+(i+2));

        if(i+2<n&&Primes.checkForPrime(i+1)&&Primes.checkForPrime(i+3))
        System.out.println((i+1)+", "+(i+3));

    }

}

}
```

```

1 import java.util.*;
2
3
4 import java.io.*;
5
6 import primespackage.Primes;
7
8
9
10 class TwinPrimes{
11
12     public static void main(String[] args) {
13
14         Scanner in=new Scanner(System.in);
15         System.out.println("Input limit to check for twin primes till:");
16         int n=in.nextInt();
17         if (n<5) return;
18         for(int i=3;i+3<n;i+=2){
19             if(Primes.checkForPrime(i)&&Primes.checkForPrime(i+2)) System.out.println(i+" "+(i+2));
20             if(i+2<n&&Primes.checkForPrime(i+1)&&Primes.checkForPrime(i+3)) System.out.println((i+1)+" "+(i+3));
21         }
22     }
23 }

```

5. 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'.

```

import java.util.NoSuchElementException;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class ExceptionHandling{

    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");
}
}

```

C:\My Computer\Local Disk G:\VIT\7th Semester\CSE1007 materials\Lab\Experiment 2\ExceptionHandling.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

result_abinash.java ExceptionHandling.java

```
9      if(r.length() != 9){
10          System.out.println("Invalid");
11          throw new IllegalArgumentException("Register Number does not contain exactly 9 characters");
12      }
13      if(n.length() != 10){
14          System.out.println("Invalid");
15          throw new IllegalArgumentException("Mobile Number does not contain exactly 10 characters");
16      }
17
18      String pattern = "^([a-zA-Z0-9_]{1,10})\\.([a-zA-Z0-9_]{1,10})\\.([a-zA-Z0-9_]{1,10})\\.([a-zA-Z0-9_]{1,10})\\.([a-zA-Z0-9_]{1,10})$";
19      Pattern a = Pattern.compile(pattern);
20      Matcher m1 = a.matcher(r);
21      if(!m1.find()){
22          throw new IllegalArgumentException("Invalid");
23      }
24
25      String pattern2 = "^([0-9]{10})$";
26      Pattern b = Pattern.compile(pattern2);
27      Matcher m2 = b.matcher(n);
28      if(!m2.find()){
29          throw new IllegalArgumentException("Invalid");
30      }
31
32      public static void main(String[] args) {
33          Scanner sc = new Scanner(System.in);
34          String reg = sc.next();
35          String no = sc.next();
36          sc.close();
37          validate(reg, no);
38          System.out.println("Valid");
39      }
40
41  }
```

Java source file length: 1,462 lines: 42 Ln: 1 Col: 1 Sel: 0 | 0

Desktop antmatter 19:40

No new notifications

Tablet mode Network All settings Airplane mode Location Focus assist Bluetooth Project Connect Mobile hotspot Night light Battery saver Screen snip