School of Information Technology & Engineering

Winter Semester 2021-2022

ITA5004 – Object Oriented Programming using JAVA

DIGITAL ASSIGNMENT-03

NAME:PRIYADHARSHINI.R

REGNO:21MCA0032

SLOT:L3+L4

1. Create a thread class to print a multiplication table for any given number 'i'. Make the main thread produce a multiplication table for any three numbers by creating 3 threads and demonstrating it.

CODE:

```
import java.io.*;
class A extends Thread
{
public void run()
{
for (int i = 1; i <= 7; i++)
{
System.out.println(i + "*"+7+ "="+(i * 7));
}
System.out.println("END OF THE 1st THREAD");
}
}
class B extends Thread
{
public void run()
{
for (int j = 1; j <= 9; j++)
{
System.out.println(j + "*" +9+ "=" +(j * 9));
```

```
}
System.out.println("END OF THE 2st THREAD");
}

}
class C extends Thread
{
public void run()
{
for (int k = 1; k <= 12; k++)
{
System.out.println(k + "*" +12+ "=" +(k * 12));
}
System.out.println("END OF THE 3st THREAD");
}
}
public class Ans41
{
public static void main(String args[])throws IOException
{
A ThreadA=new A();
B ThreadB=new B();
C ThreadC=new C();
ThreadA.setPriority(Thread.MAX_PRIORITY);
ThreadB.setPriority(Thread.NORM_PRIORITY);
ThreadC.setPriority(Thread.MIN_PRIORITY);
ThreadA.start();
ThreadB.start(); ThreadC.start();
}
}
```

OUTPUT:

```
C:\Users\PRIYA\Downloads>javac Ans41.java

C:\Users\PRIYA\Downloads>java Ans41
1*12=12
1*9=9
1*7=7
2*9=18
2*12=24
3*12=36
4*12=48
5*12=60
6*12=72
7*12=84
8*12=96
9*12=108
10*12=120
11*12=132
12*12=144
END OF THE 3st THREAD
3*9=27
2*7=14
3*7=21
4*9=36
5*9=45
4*7=28
5*7=35
6*7=42
```

```
4*12=48
5*12=60
6*12=72
7*12=84
8*12=96
9*12=108
10*12=120
11*12=132
12*12=144
END OF THE 3st THREAD
3*9=27
2*7=14
3*7=21
4*9=36
5*9=45
4*7=28
5*7=35
6*7=42
7*7=49
6*9=54
END OF THE 1st THREAD
7*9=63
8*9=72
9*9=81
END OF THE 2st THREAD

C:\Users\PRIYA\Downloads>
```

2. Write a java program using threads to compute the first 25 prime numbers, and to compute the first 50 Fibonacci numbers. Set the priority of the thread that computes the Fibonacci number to 8 and the other to 5. After calculating 30 Fibonacci numbers, make that thread sleep and take up prime number computation. After computing the 25 prime numbers continue the Fibonacci number computing.

CODE:

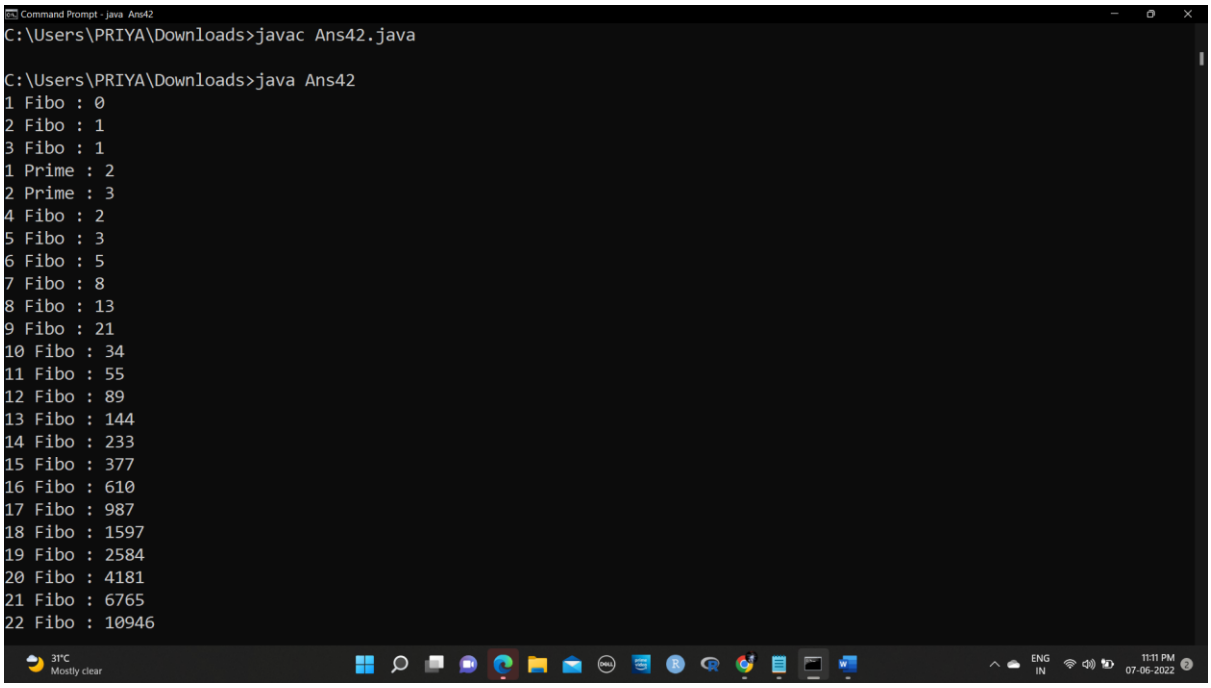
```
import java.io.*;
```

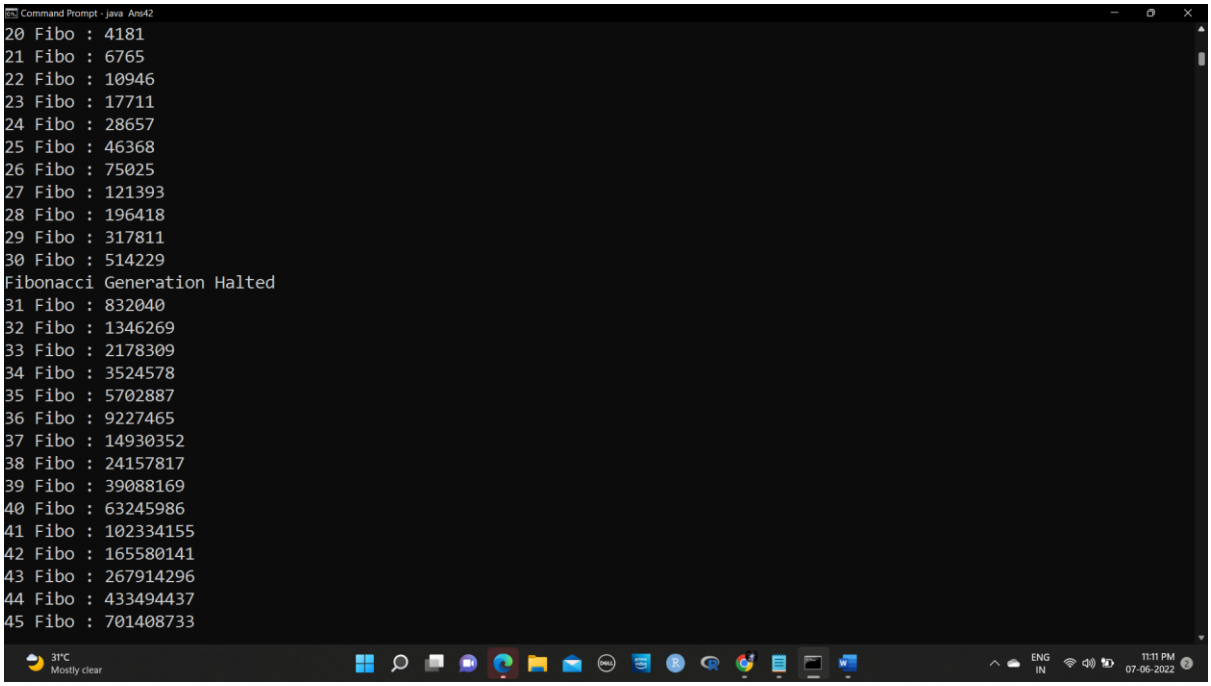
```
class Fibo{
private int n=1,a=-1,b=1,c;
synchronized void disp(){

for(int i=0;i<=45;i++){
if(n==31)
try{
System.out.println("Fibonacci Generation Halted");
Thread.sleep(4000);
}catch(InterruptedException e){
System.out.println("Caught interrupted exception");
}
c=a+b;
System.out.println(n+" Fibo : "+c);
a=b;
b=c;
n++;
}
}
}
class Prime{
int n=1;
boolean isPrime=true;
synchronized void disp(){
for(int i=2;;i++){
for(int j=2;j<=i/2;j++){
if((i%j)==0){
isPrime = false;
break;
}
```

```
}
if(isPrime){
System.out.println(n+" Prime : "+i); n++;
if(n==25){ break;
}
}
}
}
}

class PrimeThread implements Runnable{ Thread t;
Prime p1; PrimeThread(){
t=new Thread(this);
t.setPriority(Thread.NORM_PRIORITY);
t.start();
}
public void run(){
p1=new Prime();
p1.disp();
}
}

class FiboThread implements Runnable
{
Thread t2;
Fibo f;
FiboThread(){
t2=new Thread(this);
t2.setPriority(7);
```

```
t2.start();
}
public void run(){
f=new Fibo();
f.disp();
}
}

public class Ans42
{

public static void main(String args[])
{
FiboThread ft=new FiboThread();
PrimeThread pt=new PrimeThread();
}
}
```

OUTPUT:





3. Demonstrate the synchronized methods in threads by an example program. (Hint- Bank account with operations such as Deposit(), Withdraw() and checkBalance())

CODE:

```
import java.io.*;

class BankAccount

{

private int balance;

public BankAccount(int bal)

{

balance = bal;

}

public BankAccount()

{

this(0);

}

public synchronized int checkBalance()

{

return balance;
```

```
}
public synchronized void deposit(int amt)
{
int temp = balance;
temp = temp + amt;
try {
Thread.sleep(300);
}
catch (InterruptedException ie)
{
System.err.println(ie.getMessage());
}
System.out.println("After deposit balance is:" + temp);
balance = temp;
notify();
}

public synchronized void withdraw(int amt)
{

while (balance < amt)
{
try
{
wait();
}
catch (InterruptedException ie)
{
System.err.println(ie.getMessage());
}
```

```
}

int temp = balance;
temp = temp - amt;
try {
Thread.sleep(200);
}
catch (InterruptedException ie)
{
System.err.println(ie.getMessage());
}
System.out.println("After withdrawl balance is:" + temp); balance = temp;
}
}

public class Ans33
{

public static void main(String args[])
{
BankAccount obj=new BankAccount(1000); obj.deposit(3000);
obj.withdraw(500);
}
}
```

OUTPUT:

```
C:\Users\PRIYA>cd downloads

C:\Users\PRIYA\Downloads>javac Ans33.java

C:\Users\PRIYA\Downloads>java Ans33
After deposit balance is:4000
After withdrawl balance is:3500

C:\Users\PRIYA\Downloads>
```

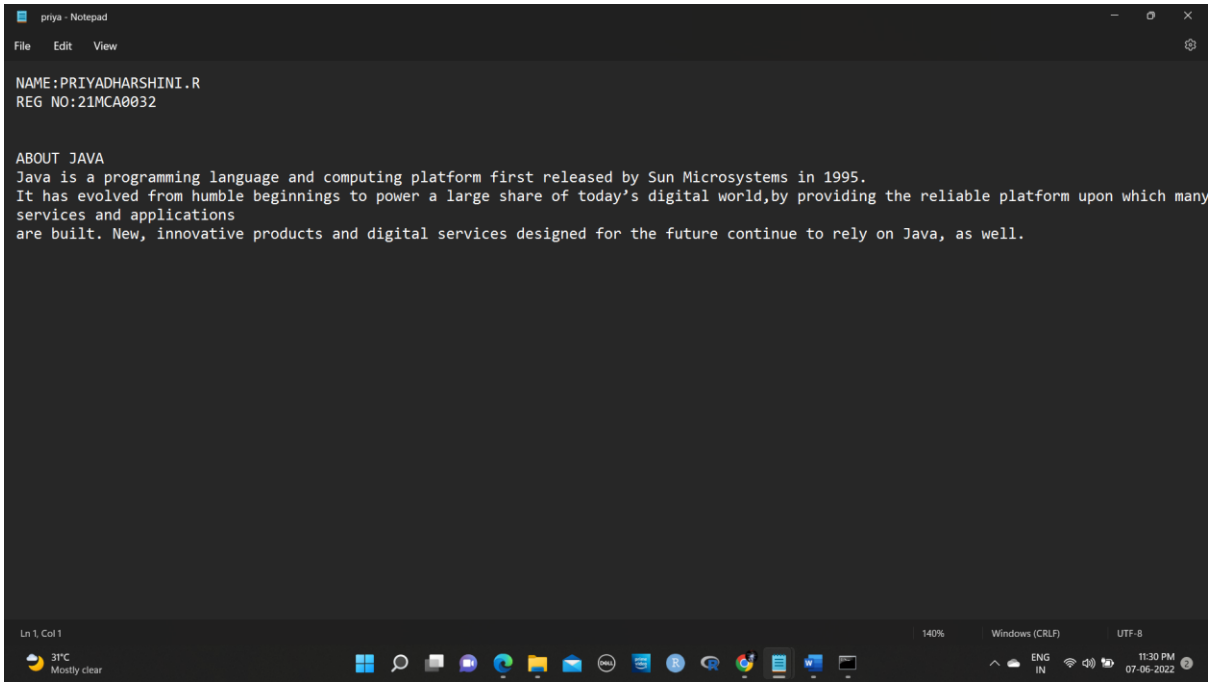
4. Write a program to create a text file and copy the contents of this file to another file and then display it.

CODE:

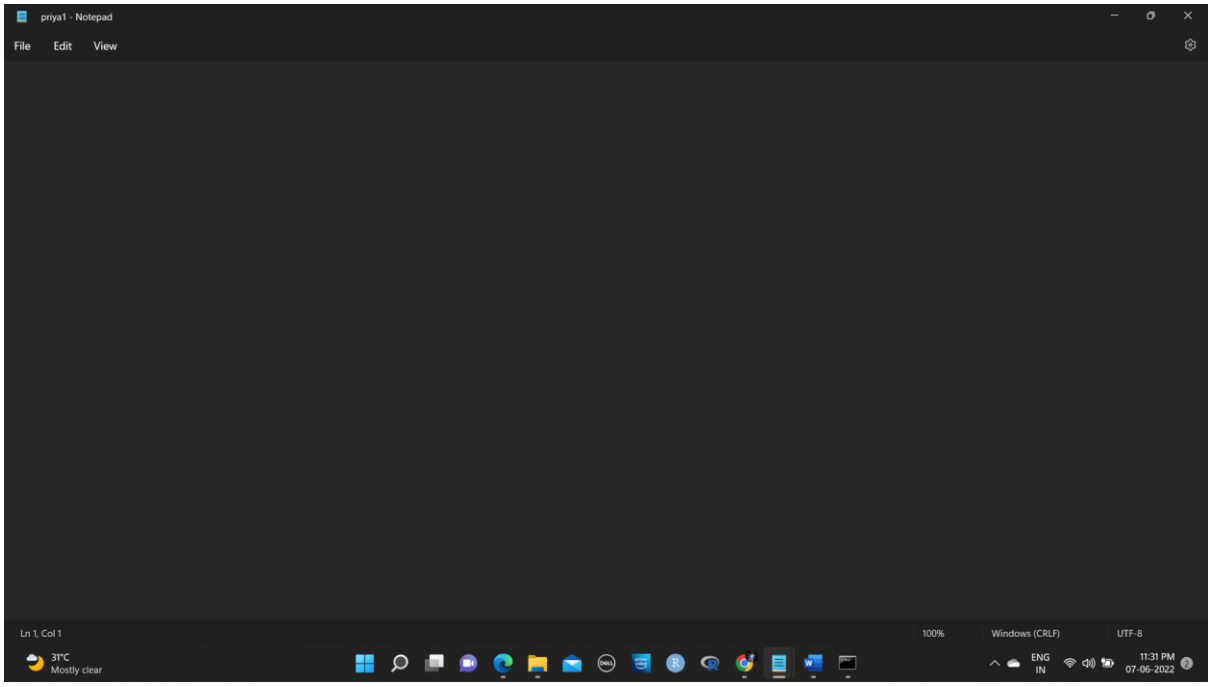
OUTPUT:

```
C:\Users\PRIYA\Downloads>javac Ans44.java

C:\Users\PRIYA\Downloads>java Ans44
Enter the source filename from where you have to copy :
priya.txt
Enter the destination filename where you have to paste :
priya1.txt
File Copied

C:\Users\PRIYA\Downloads>
```

Priya.txt

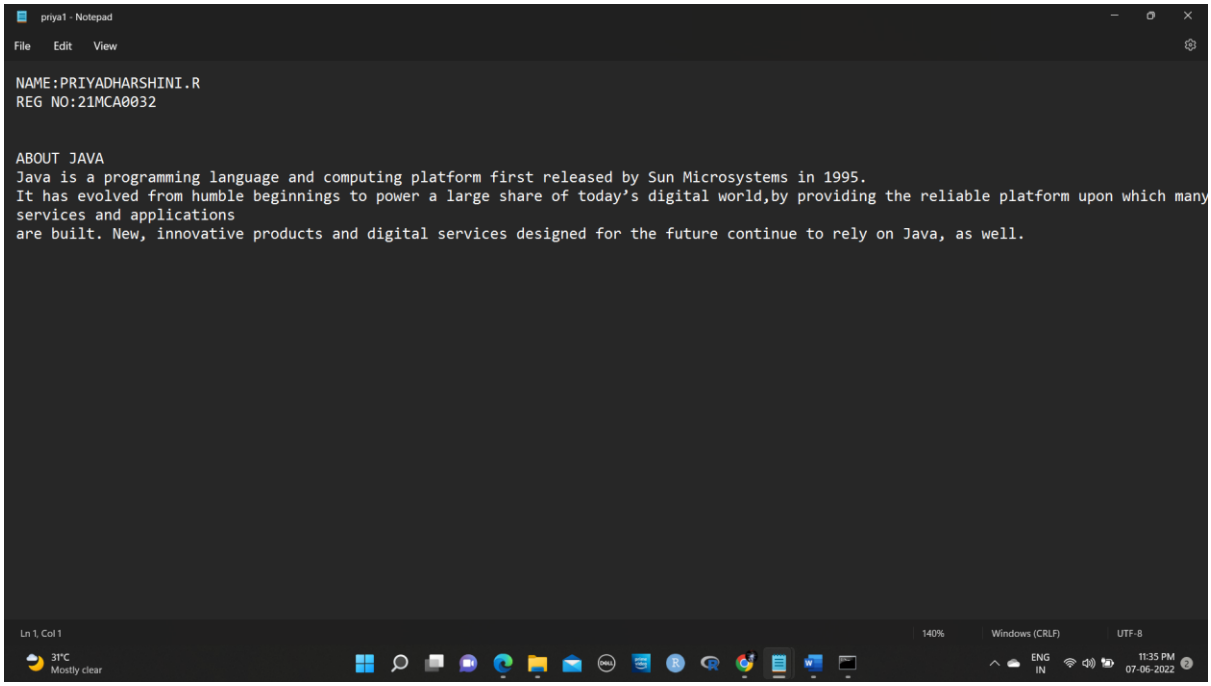


Priya1.txt



AFTER COPYING THE CONTENT:

Priya.txt



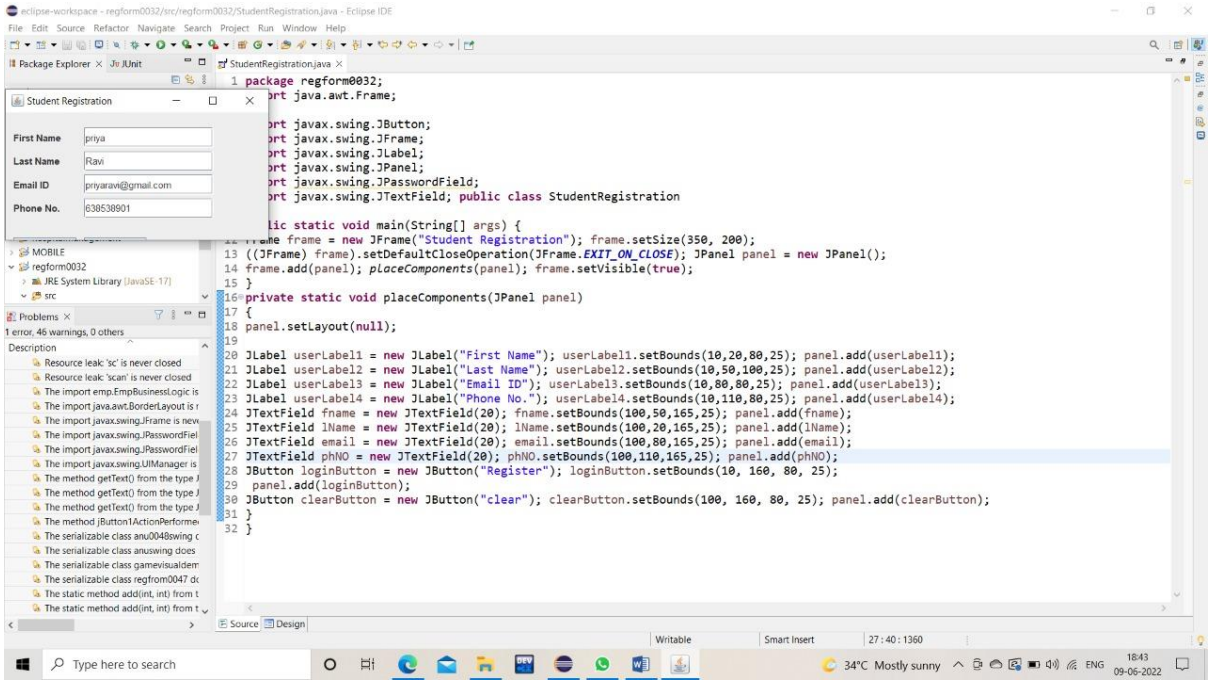
5. Create any registration form (of your choice) using appropriate AWT/SWING controls (such as Label, TextField, Checkbox, Lists, Button etc). When you click on the Button, display all the contents of the form in TextArea, which is part of the form and placed at the bottom of the form.

CODE:

```
package regform0032;

import java.awt.Frame;


import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField; public class StudentRegistration

{

public static void main(String[] args) {

Frame frame = new JFrame("Student Registration"); frame.setSize(350, 200);

((JFrame) frame).setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); JPanel panel = new JPanel();
```

```
frame.add(panel); placeComponents(panel); frame.setVisible(true);

}

private static void placeComponents(JPanel panel)

{

panel.setLayout(null);


JLabel userLabel1 = new JLabel("First Name"); userLabel1.setBounds(10,20,80,25);
panel.add(userLabel1);

JLabel userLabel2 = new JLabel("Last Name"); userLabel2.setBounds(10,50,100,25);
panel.add(userLabel2);

JLabel userLabel3 = new JLabel("Email ID"); userLabel3.setBounds(10,80,80,25);
panel.add(userLabel3);

JLabel userLabel4 = new JLabel("Phone No."); userLabel4.setBounds(10,110,80,25);
panel.add(userLabel4);

JTextField fname = new JTextField(20); fname.setBounds(100,50,165,25);
panel.add(fname);

JTextField lName = new JTextField(20); lName.setBounds(100,20,165,25);
panel.add(lName);

JTextField email = new JTextField(20); email.setBounds(100,80,165,25); panel.add(email);

JTextField phNO = new JTextField(20); phNO.setBounds(100,110,165,25);
panel.add(phNO);

JButton loginButton = new JButton("Register"); loginButton.setBounds(10, 160, 80, 25);

 panel.add(loginButton);

JButton clearButton = new JButton("clear"); clearButton.setBounds(100, 160, 80, 25);
panel.add(clearButton);

}

}
```

OUTPUT:



6. Demonstrate JDBC for the above question no.5. (Hint- the form data should be saved /retrieved using a database)