OOPS PRACTICAL 10

Name: Devasy Patel

Roll No: 20BCE057

Course Code: 2CS302

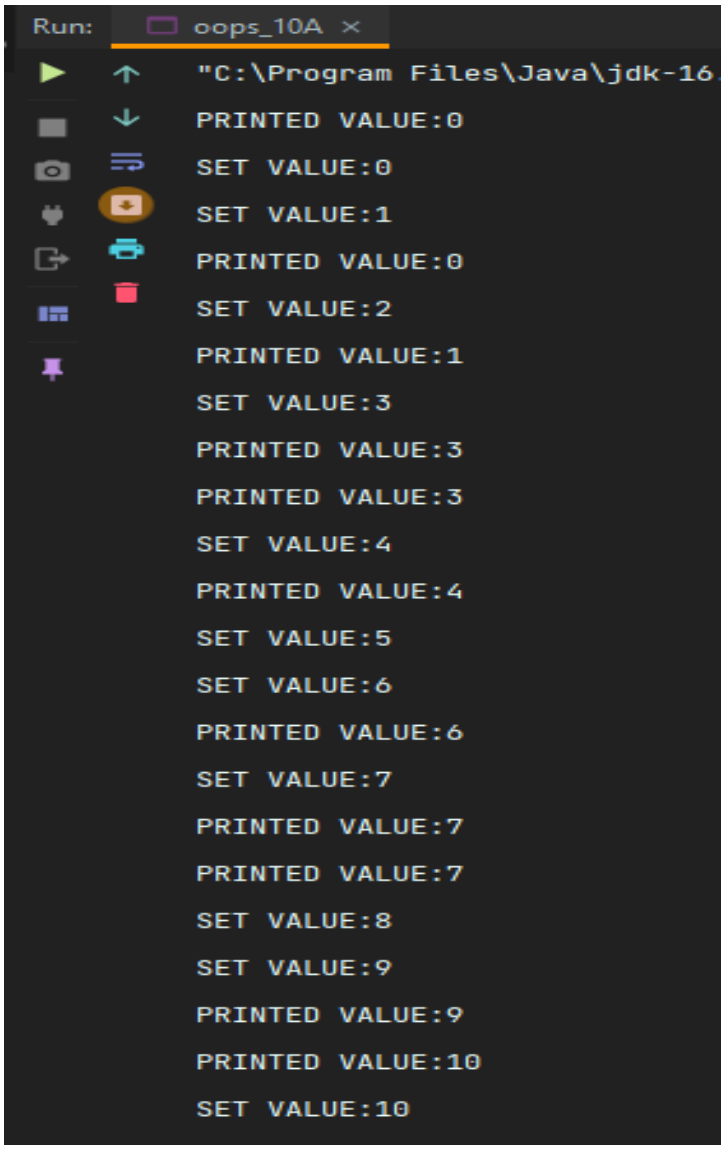
Course Name: Object Oriented Programming

Practical 10A

```
package thread;
class Storage{
    int n;
    public void getN() {
        System.out.println("PRINTED VALUE:" +n);
    }
    public void setN(int n) {
        this.n = n;
        System.out.println("SET VALUE:"+n);
    }
}
class Counter extends  Thread{
    Storage s;
    Counter(Storage c){
        this.s=c;
        Thread t1=new Thread(this);
        t1.start();  //this starts the thread by calling run method
    }
    public void run() {
        for(int i=0;i<=10;i++){
            try{
                Thread.sleep(1000);
            }catch(InterruptedException e){
                System.out.println(e);
            }
            s.setN(i);
        }
    }
}
class Printer extends Thread{
    Storage s;
    Printer(Storage p){
        this.s=p;
        Thread t2=new Thread(this);
        t2.start();
    }
    public void run() {
        for(int i=0;i<=10;i++){
            try{
                Thread.sleep(1000);
            }catch(InterruptedException e){
                System.out.println(e);
            }
            s.getN();
        }
    }
}
public class oops_10A {
    public static void main(String[] args) {
        Storage s=new Storage();
```

```
        Counter c= new Counter(s);
        Printer p=new Printer(s);
    }
}
```

OUTPUT



THEORETICAL PRINCIPLES USED:

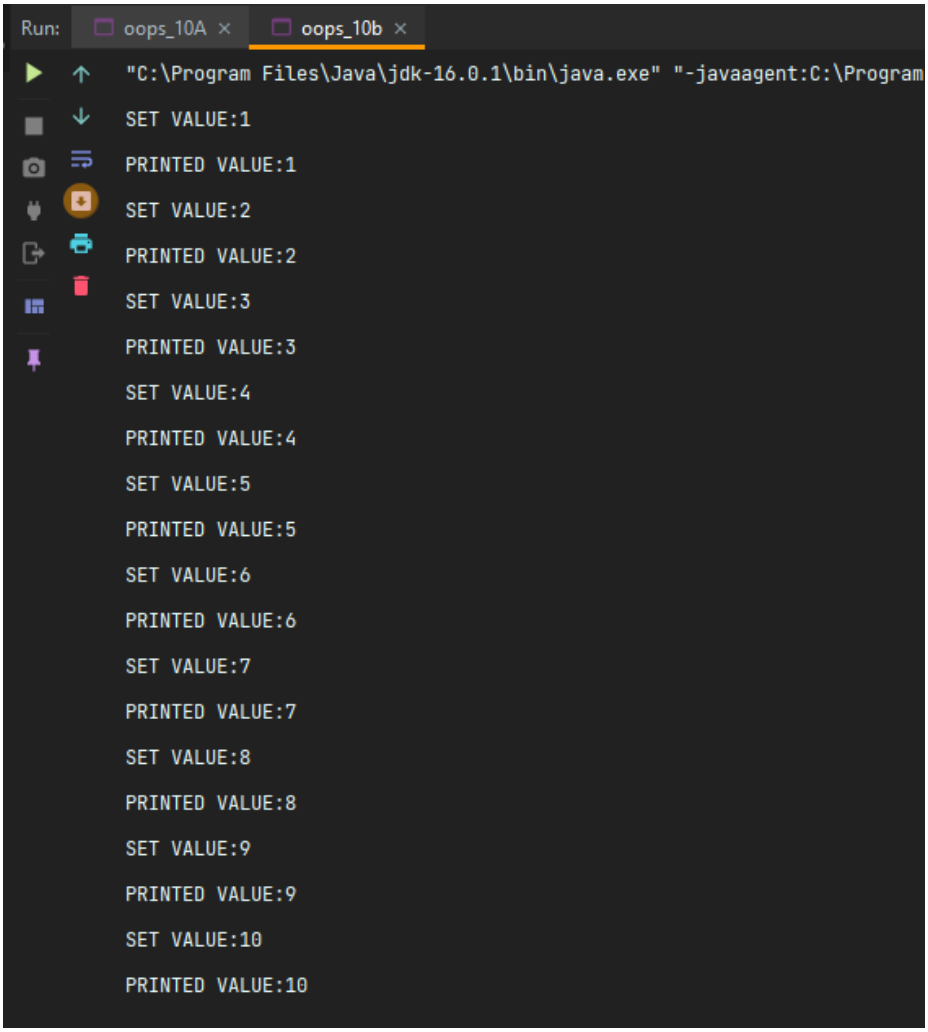
After performing this practical we will be able to use concepts of multithreading and file to perform the given task.

Practical 10B

```
package thread;
class Storage1{
    int n;
    boolean vs=true;
    synchronized public void getN() {
        while(vs){
            try{
                wait();
            }catch(InterruptedException e){
                System.out.println(e);
            }
        }
        System.out.println("PRINTED VALUE:" +n);
        vs=true;
        notify();
    }
   synchronized public void setN() {
        while(!vs){
            try{
                wait();
            }catch(InterruptedException e){
                System.out.println(e);
            }
        }
        n++;
        System.out.println("SET VALUE:"+n);
        vs=false;
        notify();
    }
}
class Counter1 implements Runnable{
    Storage1 s;
    Counter1(Storage1 c){
        this.s=c;
        Thread t1=new Thread(this);
        t1.start();  //this starts the thread by calling run method
    }
   synchronized public void run() {
        for(int i=0;i<10;i++){
            try{
                Thread.sleep(1000);
            }catch(InterruptedException e){
                System.out.println(e);
            }
            s.setN();
        }
    }
}
class Printer1 implements Runnable{
    Storage1 s;
    Printer1(Storage1 p){
        this.s=p;
        Thread t2=new Thread(this);
        t2.start();
    }
    public void run() {
        for(int i=0;i<10;i++){
            try{
                Thread.sleep(1000);
            }catch(InterruptedException e){
                System.out.println(e);
            }
            s.getN();
        }
    }
```

```
}
public class oops_10b {
    public static void main(String[] args) {
        Storage1 s=new Storage1();
        Counter1 c= new Counter1(s);
        Printer1 p=new Printer1(s);
    }
}
```

OUTPUT



THEORETICAL PRINCIPLES USED:

After performing this practical we will be able to use concepts of multithreading and file to perform the given task.

Practical 10C

```
package thread;
import java.util.*;
import java.io.*;
class ScanFile implements Runnable {
    String str;
    Thread t;
    ScanFile(String str) {
        this.str = str;
        t = new Thread(this);
        t.start();
    }
public void run() {
        try {
            File f = new File("./file1.txt");
            Scanner sc = new Scanner(f);
            boolean flag = true;
            while (sc.hasNext()) {
                if (str.equals(sc.next())) {
                    System.out.println("Word \"" + str + "\" is there in the
file");
                    flag = false;
                    break;
                }
            }
            if (flag) {
                System.out.println("Word \"" + str + "\" is not present in the
file");
            }
        }catch (Exception e) {
            System.out.println("File not found");
        }
    }
}
public class oops_10c {
    public static void main(String[] args) {
        try{
            new ScanFile(args[0]);
            new ScanFile(args[1]);
            new ScanFile(args[2]);
            new ScanFile(args[3]);
        }catch(ArrayIndexOutOfBoundsException a){
            System.out.println(a);
        }
    }
}
```

OUTPUT

```
C:\Users\Hp\IdeaProjects\FirstjavaPrg\src>javac thread/oops_10c.java


C:\Users\Hp\IdeaProjects\FirstjavaPrg\src>java thread/oops_10c hello Nirma student Aryan
Word "Nirma" is there in the file
Word "hello" is there in the file
Word "student" is not present in the file
Word "Aryan" is not present in the file
```

THEORETICAL PRINCIPLES USED:

After performing this practical we will be able to use concepts of multithreading and file to perform the given task.

Practical 10D

```
package thread;
import java.io.*;
class ScanFile2 implements Runnable {
    String fname;
    Thread t;

    ScanFile2(String name) {
        this.fname = name;
        t = new Thread(this);
        t.start();
    }
    public void run() {
        try {
            File f = new File(fname);
            BufferedReader br = new BufferedReader(new FileReader(f));
            String st;
            while ((st = br.readLine()) != null) {
                System.out.println(fname + ": " + st);
            }
        } catch (FileNotFoundException e) {
            System.out.println("File not found");
        } catch (IOException e) {
            System.out.println(e);
        }
    }
}
public class oops_10d {
    public static void main(String[] args) {
        try{
            new ScanFile2(args[0]);
            new ScanFile2(args[1]);
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Enter file names");
        }
    }
}
```

OUTPUT

```
C:\Users\Hp\IdeaProjects\FirstjavaPrg\src>javac thread/oops_10d.java


C:\Users\Hp\IdeaProjects\FirstjavaPrg\src>java thread/oops_10d file1.txt file2.txt
file2.txt: cmd
file2.txt: intelij
file2.txt: ide
file2.txt: java
file1.txt: hello
file1.txt: Nirma
file1.txt: University
```

THEORETICAL PRINCIPLES USED:

After performing this practical we will be able to use concepts of multithreading and file to perform the given task.

Practical 10E

```
import java.io.*;
class ScanFile3 implements Runnable {
    Thread t;
    String str;

    ScanFile3(String str, int priority) {
        this.str = str;
        t = new Thread(this);
        t.setPriority(priority);
        t.start();
    }
    public void run() {
        try {
            File file = new File("file1.txt");
            FileReader fr = new FileReader(file);
            BufferedReader br = new BufferedReader(fr);
            StringBuffer sb = new StringBuffer();
            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(str + line);
                t.sleep(1000);
            }
            br.close();
        } catch (IOException e) {
            System.out.println("Error reading the file");
```

```
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}
public class oops_10e {
    public static void main(String[] args) {
        Thread.currentThread().setPriority(Thread.MAX_PRIORITY);
        new ScanFile3("Priority-3:\t", 3);
        new ScanFile3("Priority-7:\t", 7);
        System.out.println("Hello");
    }
}
```

OUTPUT

```
C:\Users\Hp\IdeaProjects\FirstjavaPrg\src>javac thread/oops_10e.java


C:\Users\Hp\IdeaProjects\FirstjavaPrg\src>java thread/oops_10e file1.txt
Hello
Priority-7:     hello
Priority-3:     hello
Priority-7:     Nirma
Priority-3:     Nirma
Priority-7:     University
Priority-3:     University
```

THEORETICAL PRINCIPLES USED:

After performing this practical we will be able to use concepts of multithreading and file to perform the given task.

Practical 10F

```
package thread;
import java.io.FileWriter;
import java.io.*;
import java.util.*;
public class oops_10f {
    public static void main(String[] args) {
        while (true) {
            Scanner sc2 = new Scanner(System.in);
            int roll, age;
            String name, address;
            String scan[] = new String[4];

            System.out.println("Enter your ROLL NUMBER, NAME, AGE, ADDRESS");
            try {
                String temp = sc2.nextLine();
                scan = temp.split(" ");
                if (scan.length < 3) {
                    throw new InsufficientElements(temp);
                }
                roll = Integer.parseInt(scan[0]);
                name = scan[1];
                age = Integer.parseInt(scan[2]);
                address = scan[3];

                System.out.println("Do you want to save the data to file?(y/n)");
                char check1 = sc2.next().charAt(0);
                if (check1 == 'y') {
                    FileWriter fw = new FileWriter("data.txt", true);
                    fw.write(scan[0] + " " + scan[1] + " " + scan[2] + " " +
scan[3] + "\n");
                    fw.close();
                }
            } catch (NumberFormatException e) {
                System.out.println("Please enter only integers for age and Roll
number");
            } catch (InsufficientElements e) {
                System.out.println(e);
            } catch (IOException e) {
                System.out.println("Exception occurred while writing in file");
            } finally {
                System.out.println("Do you want to enter a new entry?(y/n)");
                char check2 = sc2.next().charAt(0);
                if (check2 == 'n') {
                    System.out.println("Thank you");
                    break;
                }
            }
        }
    }
}
class InsufficientElements extends Exception {
    String str;

    InsufficientElements(String str) {
        this.str = str;
    }
    public String toString() {
        return "Less Data is Entered";
    }
}
```

```java
//Displaying Data
import java.io.*;
public class display_data {
    public static void main(String[] args) {
        try {
            File file = new File("data.txt");
            FileReader fr = new FileReader(file);
            BufferedReader br = new BufferedReader(fr);
            StringBuffer sb = new StringBuffer();
            String line;
            System.out.println("Roll\t\tName\t\tAge\t\tAddress");
            System.out.println("-------------------------------------------------------
---------\n");
            while ((line = br.readLine()) != null) {
                String arr[] = new String[4];
                arr = line.split(" ");
                System.out.println(arr[0] + "\t\t" + arr[1] + "\t\t" + arr[2] +
"\t\t" + arr[3]);
            }
            br.close();
        } catch (IOException e) {
            System.out.println("Error reading the file");
        }
    }
}
```

OUTPUT

```
C:\Users\Hp\IdeaProjects\FirstjavaPrg\src>java thread/oops_10f
Enter your ROLL NUMBER, NAME, AGE, ADDRESS
20 Aryan 19 Mumbai
Do you want to save the data to file?(y/n)
y
Do you want to enter a new entry?(y/n)
y
Enter your ROLL NUMBER, NAME, AGE, ADDRESS
23
Less Data is Entered
Do you want to enter a new entry?(y/n)
y
Enter your ROLL NUMBER, NAME, AGE, ADDRESS
23 Khushi 10 Delhi
Do you want to save the data to file?(y/n)
y
Do you want to enter a new entry?(y/n)
n
Thank you
```

```
C:\Users\Hp\IdeaProjects\FirstjavaPrg\src>javac thread/display_data.java

C:\Users\Hp\IdeaProjects\FirstjavaPrg\src>java thread/display_data
Roll            Name            Age             Address
-------------------------------------------------------------


20              Aryan           19              Mumbai
23              Khushi          10              Delhi
```

THEORETICAL PRINCIPLES USED:

After performing this practical we will be able to use concepts of multithreading and file to perform the given task.