**Name :** Mufid Vahora

**Roll no :** 20BCE307

**Subject :** Object Oriented Programming (2CS302)

**Practical :** 10

## Practical Name : 10A

**Aim :** Implement three classes: Storage, Counter and Printer.

The Storage class should store an integer.

The Counter class should create a thread and starts counting from 0 (0, 1, 2, 3…) and stores each value in the Storage class.

The Printer Class should create a thread that keeps reading the value in the Storage class and printing it.

Write a program that creates an instance of the Storage class and set up a Counter and Printer object to operate on it.

Identify that, whether synchronization is required or not in this assignment. If yes, implement it.

## Methodology followed :

```
class Storage
{
int n;

public void set(int n)              //setting value of n
{
this.n = n;
System.out.println("Value set is "+n);
}

public int get()            //returning value of n
```

```
{
return n;
}
}

class Counter extends Thread
{
Storage s;
public Counter(Storage s)
{
this.s = s ;
Thread t1 = new Thread(this);
t1.start();
}

public void run()                    //running the values of counter
{
for(int i=0;i<=10;i++)
s.set(i);
}
}

class Printer extends Thread
{
Storage s;
public Printer(Storage s)
{
```

```
this.s = s ;
Thread t2 = new Thread(this);
t2.start();
}

public void run()                          //running the values of printer
{
for(int i=0;i<=10;i++)
System.out.println("Value is "+s.get());
}
}

public class OopPractical10a
{
public static void main(String[] args)          //running final code as asked in the question
{
Storage s = new Storage();
Counter c = new Counter(s);
Printer p = new Printer(s);
}
}
```

## Input / Output :

```
Value set is 0
Value is 0
Value set is 1
Value is 1
Value set is 2
```

Value is 2

Value set is 3

Value is 3

Value set is 4

Value is 4

Value set is 5

Value is 5

Value set is 6

Value is 6

Value set is 7

Value is 7

Value set is 8

Value is 8

Value set is 9

Value is 9

Value set is 10

Value is 10

**Conclusion :** In this practical we print the value and where it stored using storage class.

## Practical Name : 10B

**Aim :** Modify the above program i.e 10 (a) to ensure that each number is printed exactly once, by adding suitable synchronization.

## Methodology followed :

class Storage

{

```
int x;
boolean value = false;

public synchronized void set(int x) throws InterruptedException                    //for setting values
{
while(value)
{
wait();
}

this.x =x;
System.out.println("\nThread name is "+Thread.currentThread().getName() + "\n Value set is "+x);
value = true;
notify();
}

public synchronized void get() throws InterruptedException                    //for reading or getting values
{
        while(!value)
                {
                        wait();
                }
        System.out.println("Thread name is " + Thread.currentThread().getName() + "\n Value read is " + x);
        value = false;
        notify();
```

```
                    }



}

class Counter extends Thread
{
Storage s;
Counter(Storage s)
{
this.s = s;
Thread t1 = new Thread(this);
t1.start();
}

public void run()                    //running the values of counter
{
for(int i=0;i<=10;i++)
{
try
{
s.set(i);
}
catch(InterruptedException e)
{
e.printStackTrace();
}
```

```
}
}
}

class Printer extends Thread
{
Storage s;
Printer(Storage s)
{
this.s = s;
Thread t2 = new Thread(this);
t2.start();
}

public void run()                    //running the values of printer
{
for(int i=0;i<=10;i++)
{
try
{
s.get();
}
catch(InterruptedException e)
{
e.printStackTrace();
}
}
```

```
}
}

public class OopPractical10b
{
public static void main(String[] args)                    //running final code as asked in question
{
Storage s = new Storage();
Counter c = new Counter(s);
Printer p = new Printer(s);
}
}
```

## Input / Output :

```
Thread name is Thread-1
Value set is 0
Thread name is Thread-3
Value read is 0

Thread name is Thread-1
Value set is 1
Thread name is Thread-3
Value read is 1

Thread name is Thread-1
Value set is 2
Thread name is Thread-3
Value read is 2
```

Thread name is Thread-1

Value set is 3

Thread name is Thread-3

Value read is 3

Thread name is Thread-1

Value set is 4

Thread name is Thread-3

Value read is 4

Thread name is Thread-1

Value set is 5

Thread name is Thread-3

Value read is 5

Thread name is Thread-1

Value set is 6

Thread name is Thread-3

Value read is 6

Thread name is Thread-1

Value set is 7

Thread name is Thread-3

Value read is 7

Thread name is Thread-1

Value set is 8

Thread name is Thread-3

Value read is 8

Thread name is Thread-1

Value set is 9

Thread name is Thread-3

Value read is 9

Thread name is Thread-1

Value set is 10

Thread name is Thread-3

Value read is 10

**Conclusion :** In this practical we can able to display the value and from which thread the value is read.

## Practical Name : 10C

**Aim :** Write a multithreaded program that will accept 4 strings from the command line and search in a particular file for a given string and display the status of each search on the screen. Note that, all threads are operating on the same file.

## Methodology followed :

```
package Practical10;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;
```

```
class Search extends Thread{
String s;

public Search(String s) { this.s=s;
Thread t=new Thread(this); t.start();
}
public void run() {
File      f=new
File("C:\\Users\Mufid\\Documents\\Nirma University SEM-3\\OOPS\\programs\\Sample Input.txt");
try {
Scanner sc=new Scanner(f); String have=null; while(sc.hasNext()){

have=have+sc.nextLine().toLowerCase();
}
if(have.contains(s.toLowerCase()))
{
System.out.println("String "+ s +" is Present in File");
}

else {
System.out.println("String "+ s + " is Not Present in File");
}
}catch(FileNotFoundException e) {
System.out.println("FileNotFound");
}
```

```
}


}
public class OopPractical10c {
public static void main(String[] args) {
Search s1=new Search(args[0]);
Search s2=new Search(args[1]);
Search s3=new Search(args[2]);
Search s4=new Search(args[3]);
}
}
```

**Sample Input.txt**

My Name is Mufid Vahora Student of Nirma University

## Input / Output :

String Vahora is Present in File

String Nirma is Present in File

String Mufid is Present in File

String Abcd is Not Present in File

**Conclusion :** In this practical we make a file and then display each search that it present in the file or not.

## Practical Name : 10D

**Aim :** Write a Java application that will accept two filenames (text files) as command line arguments and use two threads to read contents from the two text files. Each of the threads should sleep for a random time after displaying filename with each line.

## Methodology followed :

```
package Practical10;


import java.io.File;

import java.io.FileNotFoundException;

import java.util.Random;

import java.util.Scanner;


class Readfile extends Thread{ String s;

Readfile(String s){ this.s=s;

Thread t=new Thread(this); t.start();

}

public void run() {

File      f=new  File("C:\\Users\\Mufid\\Documents\\Nirma University
SEM-3\\OOPS\\programs\\Practical10\\"+s+".txt");

try {

Scanner sc=new Scanner(f);


while(sc.hasNext()) {

System.out.println("File          name: "+s+" Content "+sc.nextLine());

}

sleep(800);

}catch(FileNotFoundException           | InterruptedException e) {

System.out.println(e);


}

}

}
```

```
public class OopPractical10d {

public static void main(String[] args) { Readfile f1=new Readfile(args[0]); Readfile f2=new Readfile(args[1]);

}

}
```

**File1.txt**

Hey ! This is Aditi

**File2.txt**

Nirma University Student

## Input / Output :

File name: File2 Content Nirma University  Student

File name: File1 Content Hey ! This is Mufid

**Conclusion :** In this practical we can write file name and it is shows the all content in the file and display it.

## Practical Name : 10E

**Aim :** Create thread t1 and t2. Set priority of main thread to 10, t1 to  normal priority +2 and t2 to normal priority -2 d) Write a Java application that will accept two filenames (text files) as command line arguments and use two threads to read and display contents from the two text files.

## Methodology followed :

```
package Practical10;

import java.io.File;
```

```
import java.io.FileNotFoundException;
import java.util.Scanner;

class Read extends Thread{
        String s;
Read(String s){
this.s=s;
Thread t=new Thread(this);
t.start();
}
public void run() {
File      f=new
File("C:\\Users\\Mufid\\Documents\\Nirma University
SEM-3\\OOPS\\programs\\Practical10\\Practical10\\"+s+".txt");
try {
Scanner sc=new Scanner(f);

while(sc.hasNext()) {
System.out.println("Filename: "+s+" Content "+sc.nextLine());
}
sleep(800);
}
catch(FileNotFoundExceptionInterruptedException e) {
System.out.println(e);

}
}
}
```

```
public class OopPractical10e {

public static void main(String[]args) { Thread.currentThread().setPriority(10); Thread t1=new Read(args[0]);

Thread t2=new Read(args[1]);



t1.setPriority(Thread.NORM_PRIORITY+2);


t1.setPriority(Thread.NORM_PRIORITY-2);




t1.start();

t2.start();




}

}
```

**File1.txt**

Hey ! This is Xyz

**File2.txt**

Nirma University Student

## Input / Output :

File name: File1 Content Hey ! This is Mufid

File name: File2 Content Nirma University Student

File name: File1 Content Hey ! This is Mufid

File name: File2 Content Nirma University Student

**Conclusion :** In this practical we can write file name and it is shows the all content in the file and display it using thread t1 and t2.

## Practical Name : 10F

**Aim :** Write a stream based program which will accept Roll Number, Name, Age and Address from user. Age and Roll-no should be numeric. Handle with built-in exception. None of the field should be blank. Handle with custom exception. Ask user, whether to write the data in the file. If answer is yes, then data is saved into a file as an object (User can write many records in the file), otherwise terminate the current program. Write another program to display all the records saved into the file

## Methodology followed :

```
import java.io.*;                    //importing whole package

import java.util.Scanner;            //importing for scanning


class Empty_Exception extends Exception          //user defined exception named
Empty_Exception

{

String str1;


Empty_Exception(String str2)

{

str1 = str2;

}
```

```
public String toString()
{
return ("Empty Value Exception : "+str1);
}
}
public class OopPractical10f
{
public static void main(String[] args) throws IOException
{
int k=1;
while(k==1)                //user can repeat this program how many times he/she wants
{
Scanner sc = new Scanner(System.in);                    //assigning sc for scanning
StringBuilder sb = new StringBuilder();
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String input;
System.out.println("Enter your roll Number(in integer only)");                //asking data from user
System.out.println("Enter your name");
System.out.println("Enter your age");
System.out.println("Enter your address");

try
{
int x;
for(int i=0; i<4; i++)
{
if(i>0)
```

```
sb.append(",");                    //seperating data by ','
                        sb.append(input = br.readLine());

                        if(i==0 || i==2)
                            x = Integer.parseInt(input);          //for roll number and age we use integer to string function

                        if(input.isEmpty())                //if field is empty we throw our user defined exception
                            throw new Empty_Exception("Field is empty....This cannot be empty");
                            if(i==3)                           //after the data is entered, we seperate by some other data by leaving a line
                            sb.append("\n");
                      }

                System.out.println("\nEntered data is:");          //we show what user has typed
                System.out.println(sb);
                System.out.println("Enter 1 if you want to write data in file : File_Sample.txt");
          //we ask if user wants to add his entered data in file or not
                int y = sc.nextInt();

                if(y==1)
                  {
                    File f = new File("C:\\Users\\Mufid\\Documnets\\Nirma University SEM-3\\OOPS\\programs\\File_Sample.txt");                    //defining the destination of file
                    FileWriter fw = new FileWriter("C:\\Users\\Mufid\\Documnets\\Nirma University SEM-3\\OOPS\\programs\\File_Sample.txt",true);
```

```
                    if(f.exists())          //if file exists in above destination, we enter the
data in it and display below message and then close the file
{
fw.write(String.valueOf(sb));
System.out.println("Written Successfully");
fw.close();
}
                    else          //else we display that file is not found
System.out.println("File not found!!!");
}

System.out.println("Enter 1 to repeat the program and any other number to exit");
//user can repeat this program again if he/she wants by entering 1
k = sc.nextInt();

}

catch(Empty_Exception e)                    //catching the thrown exception of our user
defined exception
{
System.out.println(e);
}
          catch(NumberFormatException e)                    //if roll number and age is
not an integer
{
System.out.println(e + "Entered value should be an integer");
}

}
```

```
}
}
```

## Input / Output :

```
Enter your roll Number(in integer only)
Enter your name
Enter your age
Enter your address
307
Mufid
19
vafsh fvgshb

Entered data is:
307,Mufid,19,vafsh fvgshb

Enter 1 if you want to write data in file : File_Sample.txt
1
Enter your roll Number(in integer only)
Enter your name
Enter your age
Enter your address
453
XYz
19
fgsfhdbdfagv

Entered data is:
```

453,XYz,19,fgsfhdbdfagv

Enter 1 if you want to write data in file : File_Sample.txt

0

| Roll Number | Name | Age | Address |
|---|---|---|---|
| 307 | Mufid | 19 | vafsh fvgshb |
| 453 | XYz | 19 | fgsfhdbdfagv |

**Conclusion :** In this practical we display the given data which is enter in the list.