Name: Raghav Maheshwari Date: 21/04/22
Roll NO: 53
Panel: A

JP Lab - ASSignment - 6

Aim:

Write a Java program to showcase use of multi threading and Exception Handling in Java.

Objective:

→ Apply Multithreading in Java.
→ Apply Exception Handling in Java Applications.

Platform:

Open Source Java Programming tool like Eclipse Editor/ Netbeans.

Conclusion:

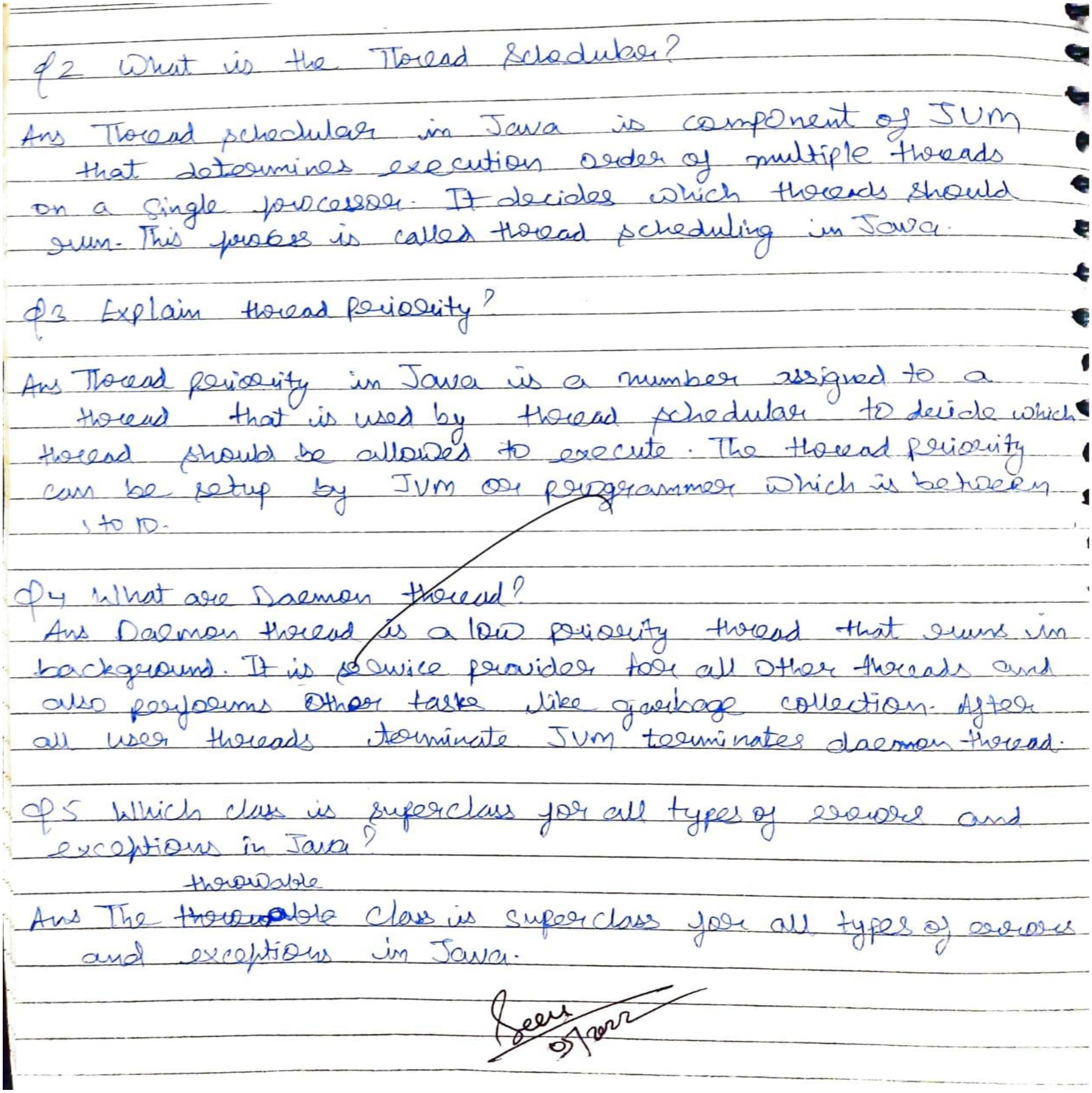
Thus studied multithreading and exception handling concepts in Java.

FAQ

Q1 What are different states in a thread lifecycle.

Ans A thread can be in these states

1. New
2. Runnable
3. Blocked
4. Waiting
5. Timed Waiting
6. Terminated.



Q2 What is the Thread Scheduler?

Ans Thread schedular in Java is component of JVM that determines execution order of multiple threads on a single processor. It decides which threads should run. This process is called thread scheduling in Java.

Q3 Explain thread priority?

Ans Thread priority in Java is a number assigned to a thread that is used by thread schedular to decide which thread should be allowed to execute. The thread priority can be setup by JVM or programmer which is between 1 to 10.

Q4 What are Daemon thread?

Ans Daemon thread is a low priority thread that runs in background. It is service provider for all other threads and also performs other taske like garbage collection. After all user threads terminate JVM terminates daemon thread.

Q5 Which class is superclass for all types of errors and exceptions in Java?

Ans The ~~throwable~~ throwable Class is superclass for all types of errors and exceptions in Java.

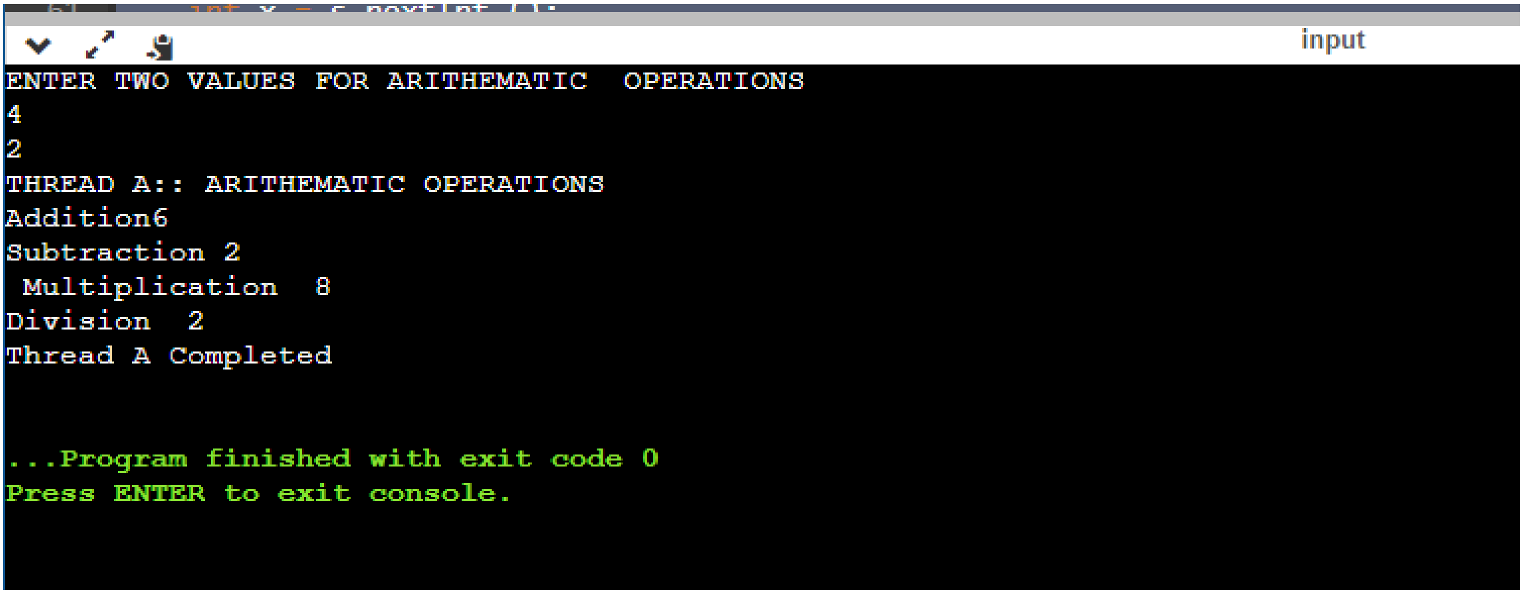
```
int i, j;
```

CODE:

```
import java.lang.*;

import java.util.*;

// Thread A( for arithematic operations)

class A1 extends Thread
{

  int i, j;
  A1 (int x, int y)
{

  i = x;

  j = y;

}

public void run ()
{


  System.out.println ("THREAD A:: ARITHEMATIC OPERATIONS");




  System.out.println ("Addition" + (i + j));

  System.out.println ("Subtraction " + (i - j));

  System.out.println (" Multiplication  " + (i * j));

  System.out.println ("Division  " + (i / j));
  System.out.println ("Thread A Completed");
  A1 a = new A1 (x, y);
  }

}


class Main
{

  public static void main (String args[])
  {

    Scanner s = new Scanner (System.in);

      System.out.println ("ENTER TWO VALUES FOR ARITHEMATIC  OPERATIONS");

    int x = s.nextInt ();

    int y = s.nextInt










    A1 a = new A1 (x, y);
```

```
        a.start ();

    }


}
```

Output:



```
ENTER TWO VALUES FOR ARITHEMATIC  OPERATIONS
4
2
THREAD A:: ARITHEMATIC OPERATIONS
Addition6
Subtraction 2
 Multiplication  8
Division  2
Thread A Completed


...Program finished with exit code 0
Press ENTER to exit console.
```

```
A1 a = new A1 (x, y);
```