Department of Computer Science & Engineering, SDMCET, Dharwad-2

AOOP Assignment Submission Report

[Submitted as part of CTA Assignment No-1]

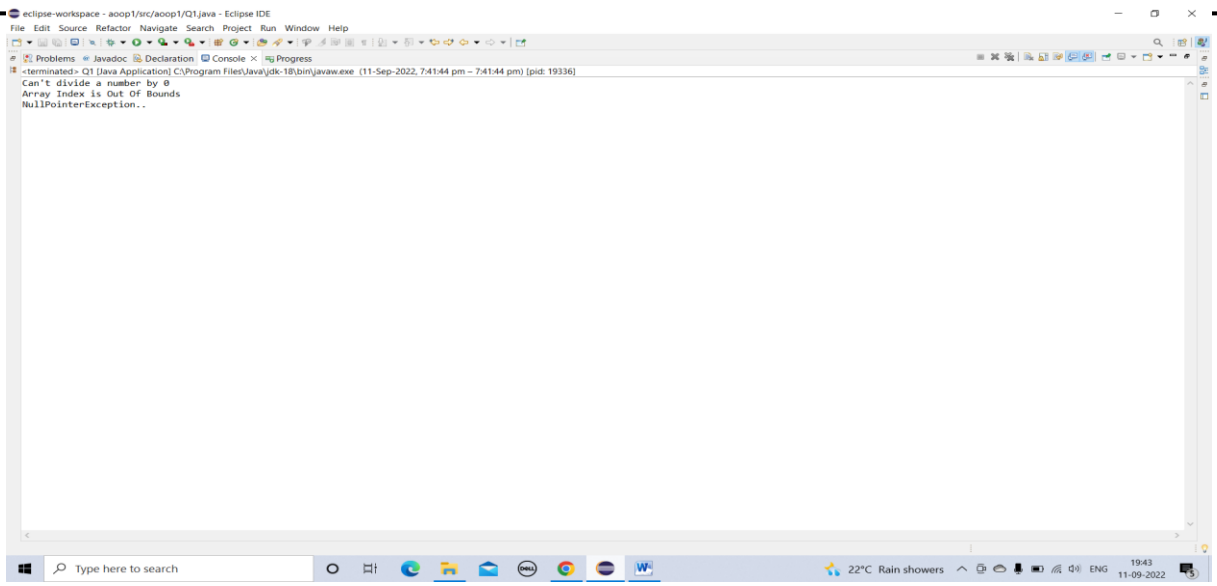
Course:	Advanced Object-Oriented Programming	Course Code:	18UCSE508
Semester:	V	Division:	A

Submitted by:

USN:	2SD20CS051	Name:	MADHUSHREE NARAYAN SARATHI

Q.1) Write a Java program to generate and handle any three built-in exceptions and display appropriate error messages.

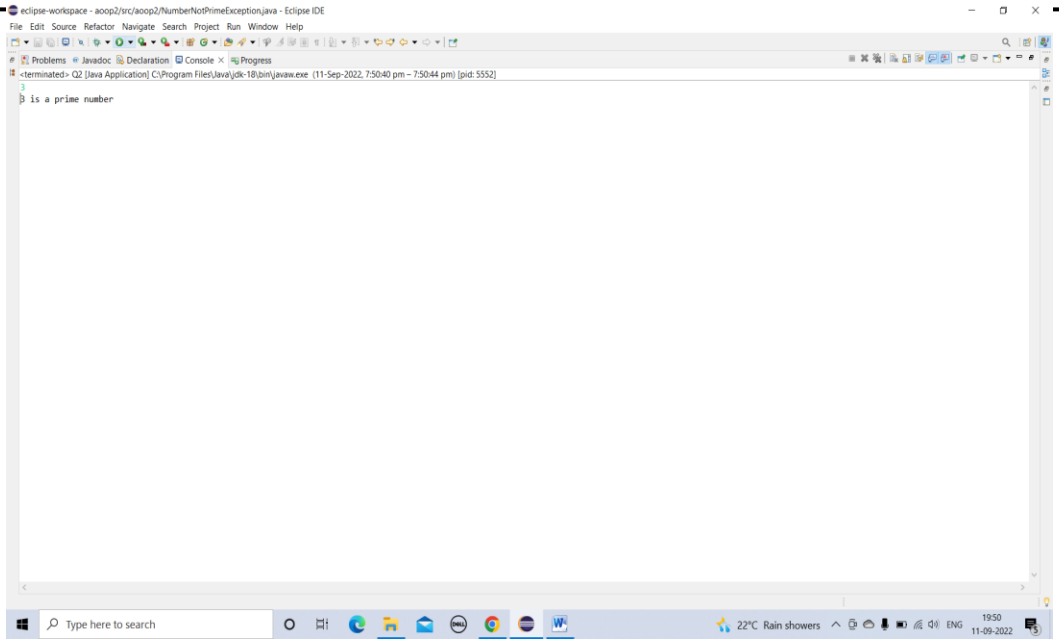
```
package aoop1;

class Q1 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		try {
	int a = 30, b = 0;
	int c = a / b; // cannot divide by zero
	System.out.println("Result = " + c);
	}
	catch (ArithmeticException e) {
	System.out.println("Can't divide a number by 0");
	}
		 try {
		int a[] = new int[5];
		a[6] = 9; // accessing 7th element in an array of
		// size 5
		}
		catch (ArrayIndexOutOfBoundsException e) {
		System.out.println("Array Index is Out Of Bounds");
		}
		 try {
		String a = null; // null value
		System.out.println(a.charAt(0));
		}
		catch (NullPointerException e) {
		System.out.println("NullPointerException..");
		}
	}

}
```

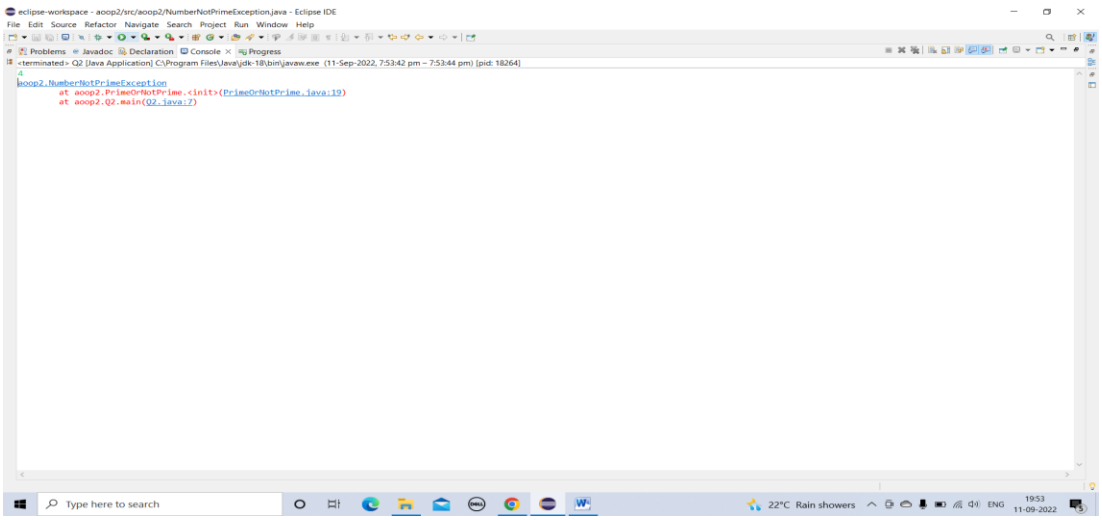


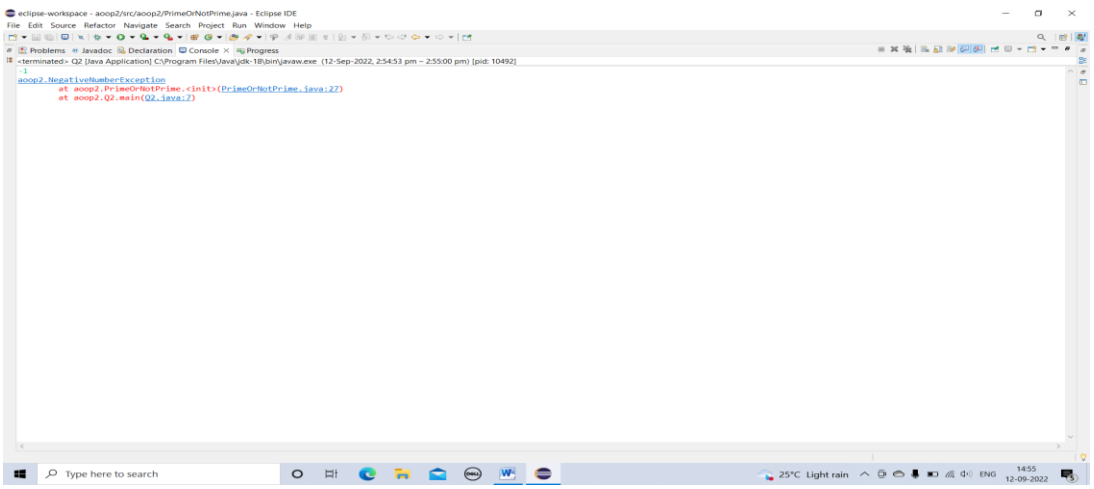
Q.2) Write a Java program to read an integer and check whether the number is prime or not. If negative number is entered, throw an exception NegativeNumberNotAllowedException and if entered number is not prime, then throw NumberNotPrimeException.

```
package aoop2;
import java.util.Scanner;
public class PrimeOrNotPrime {
	Scanner sc=new Scanner(System.in);
	//System.out.println("ENTER THE VALID NUMBER");
	int n=sc.nextInt();
	{
	try {
		if(n==0||n==1) {
			System.out.println(n+" is not a prime number");
		}
		 for(int i=2;i<=n;i++) {
			 if(n%i!=0) {
				System.out.println(n+" is a prime number");
			break;
			 }
		  else

				throw new NumberNotPrimeException("PLEASE ENTER VALID
NUMBER");
		  }
		}
	  catch(Exception e){
				e.printStackTrace();
```

```
            }
        try {
            if(n<1)
                throw new NegativeNumberException("ENTER POSITIVE NUMBER");
        }catch(Exception e1) {
            e1.printStackTrace();
        }

    }
}
package aoop2;

public class NumberNotPrimeException extends Exception {
        private String e;
        NumberNotPrimeException(String e){
            this.e=e;
        }
}
package aoop2;

public class NegativeNumberException extends Exception {
        private String e1;
        NegativeNumberException(String e1){
            this.e1=e1;
        }

}
package aoop2;

public class Q2 {

        public static void main(String[] args) {
            // TODO Auto-generated method stub
             PrimeOrNotPrime p=new PrimeOrNotPrime();
                }
            }
```


eclipse-workspace - aoop2/src/aoop2/NumberNotPrimeException.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console Progress
<terminated> Q2 [Java Application] C:\Program Files\Java\jdk-18\bin\javaw.exe (11-Sep-2022, 7:50:40 pm – 7:50:44 pm) [pid: 5552]
3
3 is a prime number
Type here to search
22°C Rain showers
ENG
19:50
11-09-2022


eclipse-workspace - aoop2/src/aoop2/NumberNotPrimeException.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console Progress
<terminated> Q2 [Java Application] C:\Program Files\Java\jdk-18\bin\javaw.exe (11-Sep-2022, 7:53:42 pm – 7:53:44 pm) [pid: 18264]
4
aoop2.NumberNotPrimeException
at aoop2.PrimeOrNotPrime.<init>(PrimeOrNotPrime.java:19)
at aoop2.Q2.main(Q2.java:7)
Type here to search
22°C Rain showers
ENG
19:53
11-09-2022


eclipse-workspace - aoop2/src/aoop2/PrimeOrNotPrime.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console Progress
<terminated> Q2 [Java Application] C:\Program Files\Java\jdk-18\bin\javaw.exe (12-Sep-2022, 2:54:53 pm – 2:55:00 pm) [pid: 10492]
-1
aoop2.NegativeNumberException
at aoop2.PrimeOrNotPrime.<init>(PrimeOrNotPrime.java:27)
at aoop2.Q2.main(Q2.java:7)
Type here to search
25°C Light rain
ENG
14:55
12-09-2022

Q.3) Write a Java program to perform the following operations:

a) Read a line of text

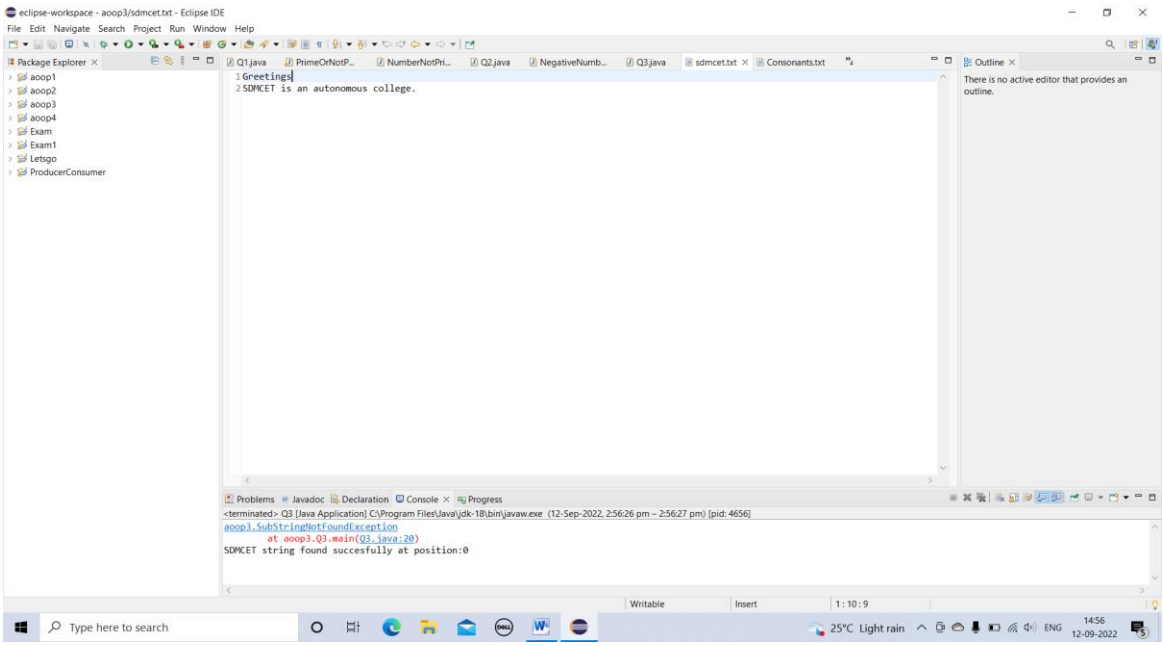
b) Search for a sub-string SDMCET (case insensitive search)

c) If found, then print success message

d) Otherwise throw an exception SubStringNotFoundException with appropriate message

```
package aoop3;
import java.io.*;
public class Q3{
   public static void main(String[] args) throws IOException{
        File file=new File("sdmcet.txt");    //creates a new file instance
        FileReader fr=new FileReader(file);   //reads the file
        BufferedReader br=new BufferedReader(fr);  //creates a buffering character input stream
        StringBuffer sb=new StringBuffer();    //constructs a string buffer with no characters
        String s1="SDMCET";
        String s2="";

        while((s2=br.readLine())!=null) {
                try {

                        if(s2.contains(s1)) {

                                System.out.println("SDMCET string found succesfully at
position:"+s2.indexOf(s1) );
                        }
                                else
                                        throw new SubStringNotFoundException("String not
found");

                        }catch(Exception e) {
                                e.printStackTrace();
                        }
                }

   }
}
package aoop3;

public class SubStringNotFoundException extends Exception{
         private String e;
         SubStringNotFoundException(String e){
                 this.e=e;
         }
```

```
}
```



Q.4) Write a Java program to perform the following operations:

a) Create a file named Alphabets.txt and insert appropriate data into it

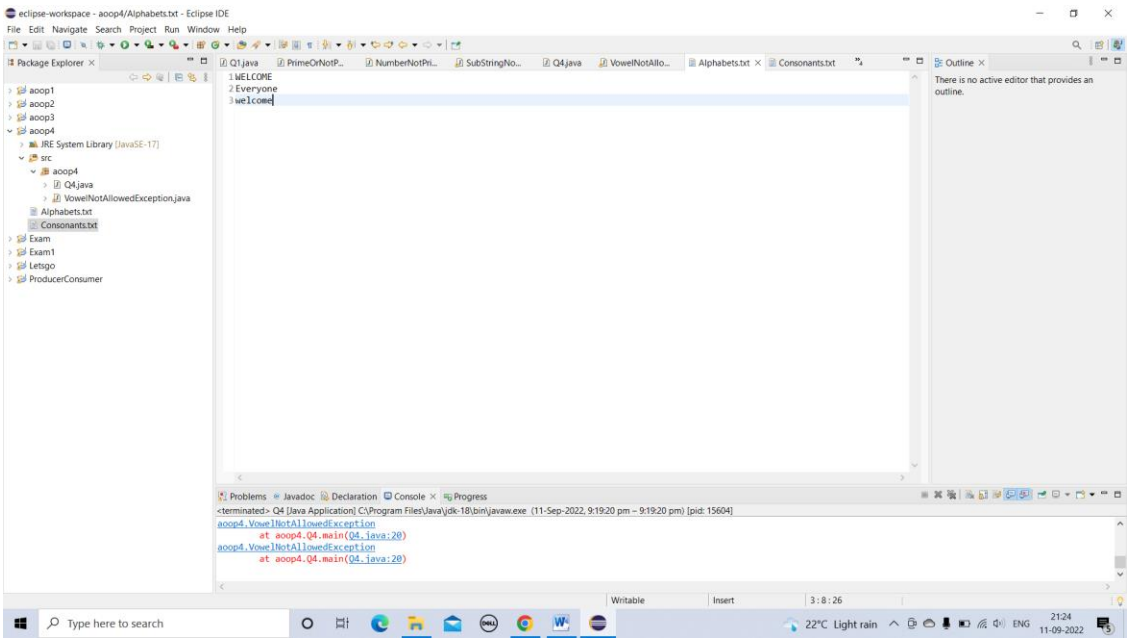
b) Read the file and copy all the consonants into another file named Consonants.txt

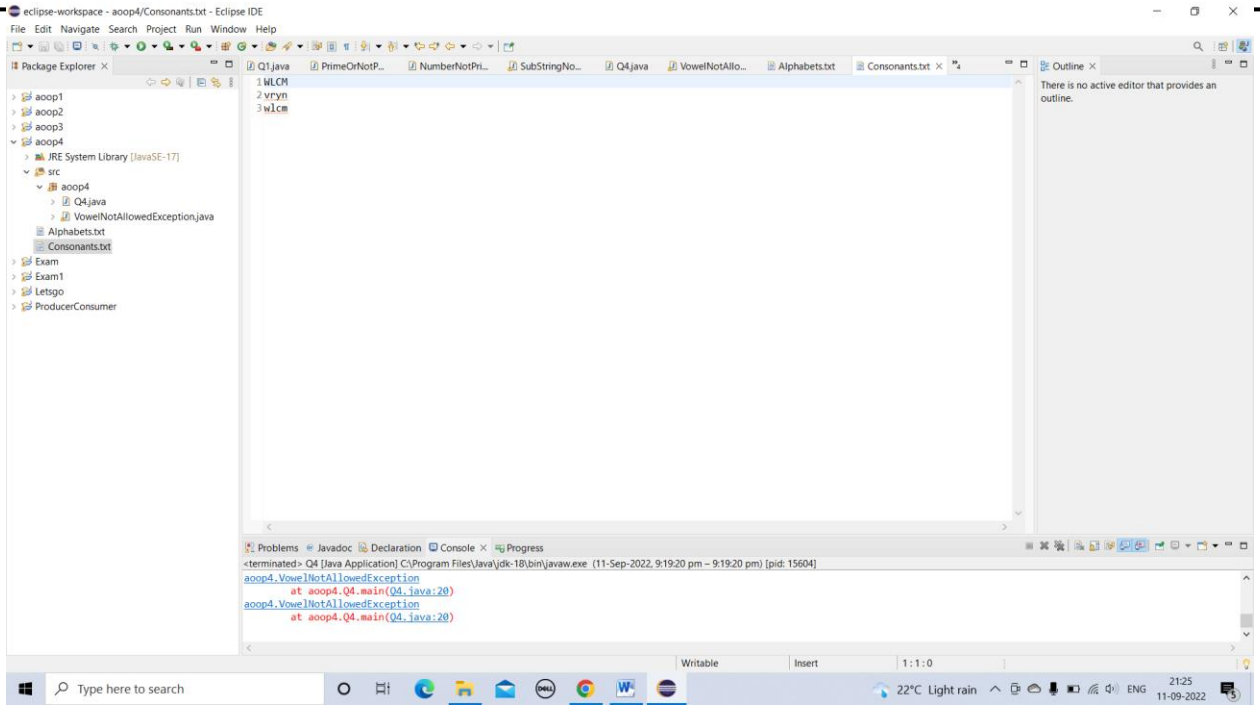
c) If vowel is encountered, throw an exception VowelNotAllowedException and continue until

end of file

```
package aoop4;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.FileNotFoundException;
public class Q4 {

        public static void main(String[] args) throws FileNotFoundException, IOException{
                // TODO Auto-generated method stub
                FileInputStream         fr =new FileInputStream ("Alphabets.txt");
                FileOutputStream fw =new FileOutputStream("Consonants.txt");
                                int ch;
                                while(( ch=fr.read())!=-1){
```

```
try {

        if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')
        {
           throw new VowelNotAllowedException("Vowels Found");
        }
        else {
             fw.write(ch);
        }
        }
         catch(Exception e) {
                e.printStackTrace();
        }
        }
                fr.close();
                fw.close();
        }

}
package aoop4;

public class VowelNotAllowedException extends Exception {
      private String e;
      VowelNotAllowedException(String e){
         this.e=e;
      }
}
```





Q.5) Write a Java program to implement the following scenario:

a) Create a file named Integers.txt and insert n-random integers into it

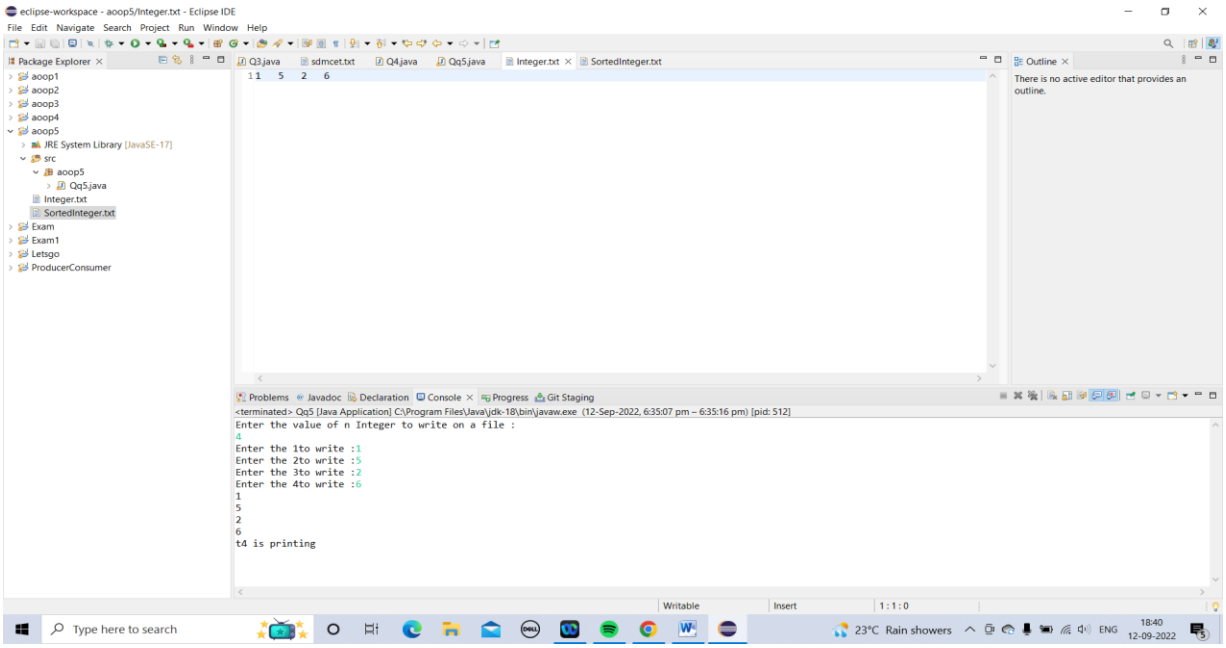
b) Create three threads T1, T2 and T3 that read n/3 integers in sequence of occurrence of numbers from the file and sort the read n/3 integers

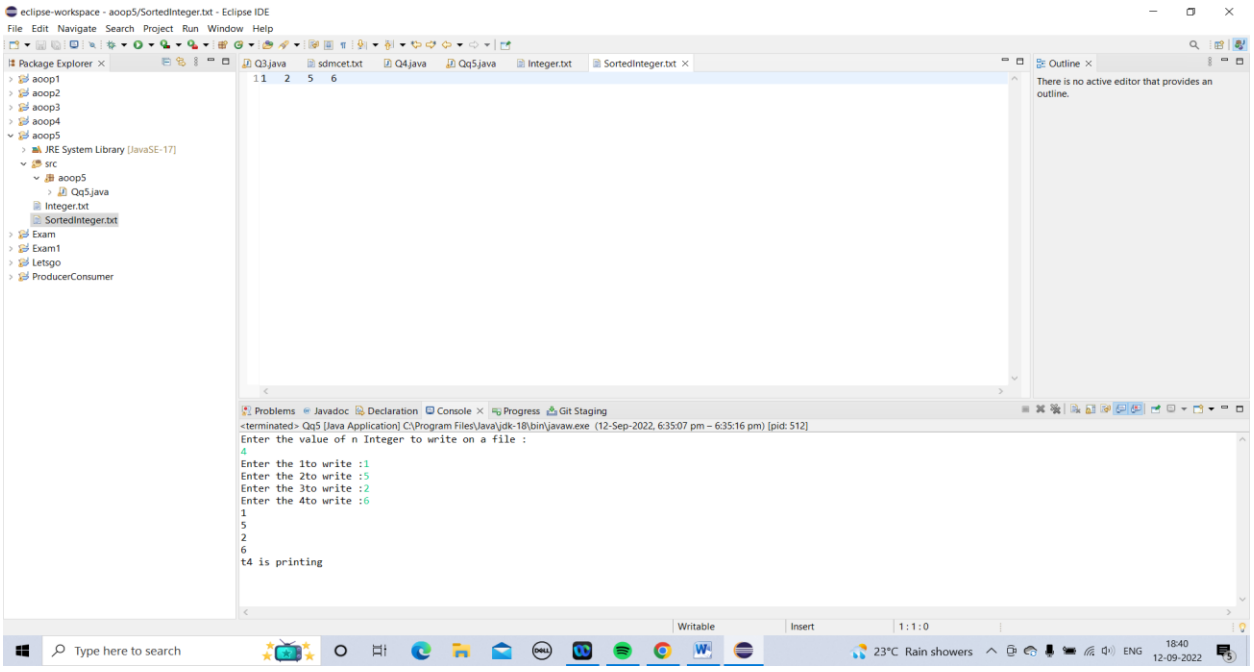
c) Thread T4 waits for all the threads T1, T2 and T3 to complete sorting, then sorts and outputs the entire list of sorted numbers to another file named SortedIntegers.txt

```
package aoop5;
import java.util.*;
import java.util.Scanner;
import java.io.*;

public class Qq5 {
	public static void main(String[] args) {
		try{
		   FileWriter w = new FileWriter("Integer.txt");
		   Scanner sc= new Scanner(System.in);
```

```
            System.out.println("Enter the value of n Integer to write on a file :");
            int n = sc.nextInt();
            for (int i = 0; i < n; i++) {
                System.out.print("Enter the " + (i + 1) +  "to write :" );
                int input = sc.nextInt();
                w.write(input + "\t");
           }w.close();
           int i=0;
           int arr[] = new int[n];
          File file = new File("Integer.txt");
          Scanner read = new Scanner(file);
          while(read.hasNext()){
               arr[i++] = Integer.valueOf(read.next());
          }

          Thread t1= new Thread(){

            public void run(){
                Arrays.sort(arr, 0, (arr.length/3));
                for (int j = 0; j < (arr.length/3); j++) {
                          System.out.println(arr[j]);
             }
             }
};
          Thread t2= new Thread(){

           public void run(){
             Arrays.sort(arr, (arr.length/3), (2*(arr.length/3)));
             for (int j =  (arr.length/3); j < (2*(arr.length/3)); j++) {
                        System.out.println(arr[j]);
              }
            }
           };
           Thread t3= new Thread(){

            public void run(){
              Arrays.sort(arr,  (2*(arr.length/3)),(n-1));
              for (int j = (2*(arr.length/3)); j < n; j++) {
```

```
                    System.out.println(arr[j]);
                  }
                 }
                };
                 Thread t4= new Thread(){

                  public void run(){
                   Arrays.sort(arr);
                   // Arrays.sort(arr,  0,n);
              StringBuilder s = new StringBuilder();
                   try{
                       FileWriter write  =new FileWriter("SortedInteger.txt");
                      System.out.println("t4 is printing");
                   for (int j = 0; j < n; j++) {
                          s.append(String.valueOf(arr[j]) + "\t");
                   }
                   write.write(s.toString());
                   write.close();
                  }catch (Exception e){
                      System.out.println(e);
                  }
                  }
                 };
                 t1.start();
                 t1.join();
                 t2.start();
                 t2.join();
                 t3.start();
                 t3.join();
                 t4.start();
          }catch(Exception e){
                    System.out.println(e);
                  }
                 }
    }
```


eclipse-workspace - aoop5/Integer.txt - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
aoop1
aoop2
aoop3
aoop4
aoop5
JRE System Library [JavaSE-17]
src
aoop5
Qq5.java
Integer.txt
SortedInteger.txt
Exam
Exam1
Letsgo
ProducerConsumer
Q3.java
sdmcet.txt
Q4.java
Qq5.java
Integer.txt
SortedInteger.txt
1 1 5 2 6
Outline
There is no active editor that provides an outline.
Problems Javadoc Declaration Console Progress Git Staging
<terminated> Qq5 [Java Application] C:\Program Files\Java\jdk-18\bin\javaw.exe (12-Sep-2022, 6:35:07 pm – 6:35:16 pm) [pid: 512]
Enter the value of n Integer to write on a file :
4
Enter the 1to write :1
Enter the 2to write :5
Enter the 3to write :2
Enter the 4to write :6
1
5
2
6
t4 is printing
Writable
Insert
1 : 1 : 0
Type here to search
23°C Rain showers
ENG
18:40
12-09-2022


eclipse-workspace - aoop5/SortedInteger.txt - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
aoop1
aoop2
aoop3
aoop4
aoop5
JRE System Library [JavaSE-17]
src
aoop5
Qq5.java
Integer.txt
SortedInteger.txt
Exam
Exam1
Letsgo
ProducerConsumer
Q3.java
sdmcet.txt
Q4.java
Qq5.java
Integer.txt
SortedInteger.txt
1 1 2 5 6
Outline
There is no active editor that provides an outline.
Problems Javadoc Declaration Console Progress Git Staging
<terminated> Qq5 [Java Application] C:\Program Files\Java\jdk-18\bin\javaw.exe (12-Sep-2022, 6:35:07 pm – 6:35:16 pm) [pid: 512]
Enter the value of n Integer to write on a file :
4
Enter the 1to write :1
Enter the 2to write :5
Enter the 3to write :2
Enter the 4to write :6
1
5
2
6
t4 is printing
Writable
Insert
1 : 1 : 0
Type here to search
23°C Rain showers
ENG
18:40
12-09-2022