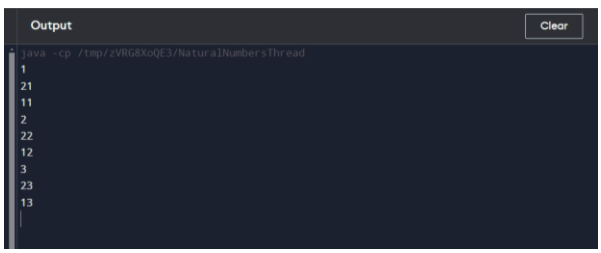
ASSIGNMENT-13

1.Write a Java program to create three threads in parallel and display the natural numbers in orders using sleep() method.

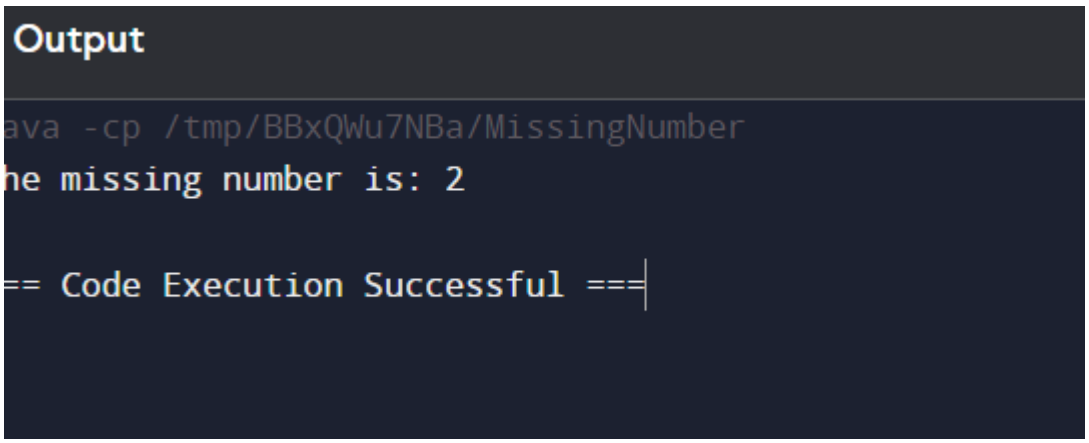
```
public class NaturalNumbersThread extends Thread {
   private int start;
   private int end;

   public NaturalNumbersThread(int start, int end) {
      this.start = start;
      this.end = end;
   }

   public void run() {
      for (int i = start; i <= end; i++) {
         System.out.println(i);
         try {
            Thread.sleep(1000); // Sleep for 1 second
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }

   public static void main(String[] args) {
      NaturalNumbersThread thread1 = new NaturalNumbersThread(1, 10);
      NaturalNumbersThread thread2 = new NaturalNumbersThread(11, 20);
      NaturalNumbersThread thread3 = new NaturalNumbersThread(21, 30);

      thread1.start();
      thread2.start();
      thread3.start();
   }
}
```



1. If n = 8, then array ‘a’ will have 7 elements in the range from 1 to 8. For example {1, 4, 5, 3, 7, 8, 6}. One number will be missing in ‘a’ (2 in this case). Write a source code to find out that missing number

   public class MissingNumber {

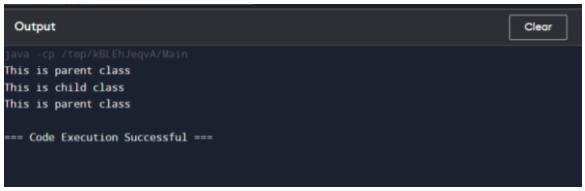
```
    public static void main(String[] args) {
        int n = 8;
        int[] a = {1, 4, 5, 3, 7, 8, 6};
        int total = n * (n + 1) / 2;
        for (int num : a) {
            total -= num;
        }
        System.out.println("The missing number is: " + total);
    }
}
```



2. Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call
   1 - method of parent class by object of parent class
   2 - method of child class by object of child class
   3 - method of parent class by object of child class

```
// Parent class
class Parent {
    public void printParent() {
        System.out.println("This is parent class");
    }
}

// Subclass inheriting from Parent
class Child extends Parent {
    public void printChild() {
        System.out.println("This is child class");
    }
}

public class Main {
    public static void main(String[] args) {
        // Create objects for parent and child classes
```

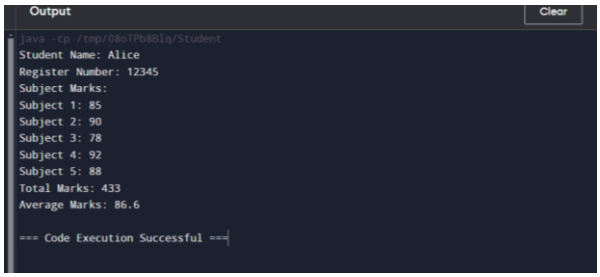
```
        Parent parentObj = new Parent();
        Child childObj = new Child();

        // Call methods using respective objects
        parentObj.printParent();
        childObj.printChild();
        childObj.printParent();
    }
}
```



3. Write a Java program to create a class Student and create constructor which assigns the values for the student details such as student name, register number, and five subject marks. Calculate the total and average of five subject marks and display the marks and average.

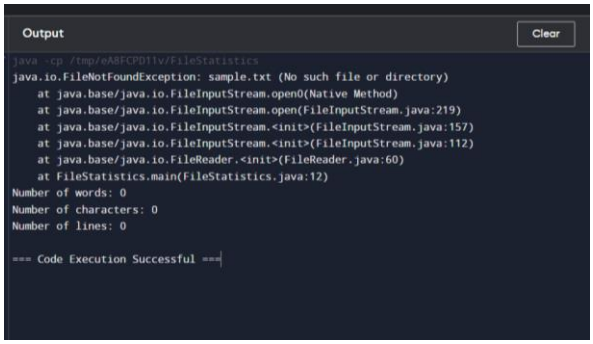
```
public class Student {
    String studentName;
    int registerNumber;
    int[] subjectMarks = new int[5];

    public Student(String name, int regNum, int[] marks) {
        studentName = name;
        registerNumber = regNum;
        subjectMarks = marks;
    }

    public void calculateTotalAndAverage() {
        int total = 0;
        for (int mark : subjectMarks) {
            total += mark;
        }
        double average = (double) total / subjectMarks.length;

        System.out.println("Student Name: " + studentName);
        System.out.println("Register Number: " + registerNumber);
        System.out.println("Subject Marks: ");
        for (int i = 0; i < subjectMarks.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + subjectMarks[i]);
        }
        System.out.println("Total Marks: " + total);
        System.out.println("Average Marks: " + average);
    }

    public static void main(String[] args) {
        int[] marks = {85, 90, 78, 92, 88};
        Student student1 = new Student("Alice", 12345, marks);
```

```
        student1.calculateTotalAndAverage();
    }
}
```



4. Generate a code to Count the Number of Words, Character and Lines from the File using Stream I/O in Java.

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class FileStatistics {
    public static void main(String[] args) {
        String filePath = "sample.txt";
        int wordCount = 0;
        int charCount = 0;
        int lineCount = 0;

        try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
            String line;
            while ((line = reader.readLine()) != null) {
                lineCount++;
                String[] words = line.trim().split("\\s+");
                wordCount += words.length;
                charCount += line.length();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        System.out.println("Number of words: " + wordCount);
        System.out.println("Number of characters: " + charCount);
        System.out.println("Number of lines: " + lineCount);
    }
}
```


Output
Clear
java -cp /tmp/eA8FCPD11v/FileStatistics
java.io.FileNotFoundException: sample.txt (No such file or directory)
at java.base/java.io.FileInputStream.open0(Native Method)
at java.base/java.io.FileInputStream.open(FileInputStream.java:219)
at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
at java.base/java.io.FileInputStream.<init>(FileInputStream.java:112)
at java.base/java.io.FileReader.<init>(FileReader.java:60)
at FileStatistics.main(FileStatistics.java:12)
Number of words: 0
Number of characters: 0
Number of lines: 0
=== Code Execution Successful ===