Lab 1

Date:

1. WAP to find greatest of three integers.

Code:

```
import java.util.*;
public class New {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter three different values\n");
        System.out.print("Enter the first value: ");
        int a=sc.nextInt();
        System.out.print("Enter the second value: ");
        int b=sc.nextInt();
        System.out.print("Enter the third value: ");
        int c=sc.nextInt();
        if (a>=b && a>=c){
            System.out.println("The greatest value is "+a);
            }
        else if(b>=a && b>=c){
            System.out.println("The greatest value is "+b);
        }
        else{
            System.out.println("The greatest value is "+c);
        }
    }
}
```

Output:

```
Output
java -cp /tmp/1K1QOFERaJ New
Enter three different values

Enter the first value: 23
Enter the second value: 78
Enter the third value: 45
The greatest value is 78
```

2. WAP to find factorial of a number using command line arguments

Code:

```
import java.util.*;
public class main
{
    static int Factorial(int n)
    {
        if(n==0 || n==1)
        {
            return 1;
        }
        else
        {
            return n* Factorial(n-1);
        }
    }
    public static void main(String[] args)
    {
        //int x = Integer.parseInt(args[0]);  //for command line argument
        int x = 7;
        int ans = Factorial(x);
        System.out.print("Factorial: "+ans);
}
}
```

Output:

```
Output
java -cp /tmp/1K1QOFERaJ main
Factorial: 5040
```

3. WAP to find the Fibonacci series using recursive and non-recursive function.

Code:

```
import java.util.*;
public class main
{
    static int Fibr(int n)
    {
        if(n==0 || n==1)
        {
            return n;
        }
        else
        {
            return Fibr(n - 1) + Fibr(n - 2);
        }
    }
    static void Fibnonr(int n)
    {
        int a = 0;
        int b = 1;
        for(int i = 0 ;i<n;i++)
        {
            System.out.print(a+" ");
            int c = a+b;
            a = b;
            b = c;
        }
    }
    public static void main(String[] args)
    {
```

```
        Scanner sc =new Scanner(System.in);
        System.out.print("How many terms do you want of the fibonacci sequence?: ");
        int n =sc.nextInt();
        System.out.println("Fibonacci series without recursion:");
        Fibnonr(n);
        System.out.println();
        System.out.println("Fibonacci series with recursion:");
        for (int i = 0; i < n; i++)
        {
            System.out.print(Fibr(i) + " ");
      }
    }
}
```

Output:

```
Output
java -cp /tmp/1K1QOFERaJ Project6
How many terms do you want of the fibonacci sequence?: 5
Fibonacci series without recursion:
0 1 1 2 3
Fibonacci series with recursion:
0 1 1 2 3
```

LAB 2

Date:

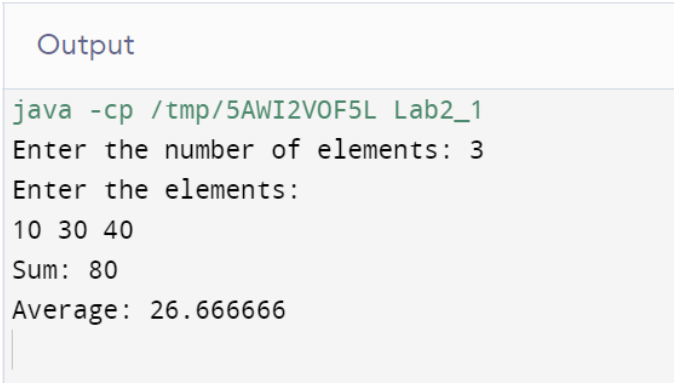
4. WAP to read a set of numbers in an array & to find the sum and average of them.

Code:

```
import java.util.Scanner;
public class Lab2_1
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = sc.nextInt();
        int arr[] = new int[n];
        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++)
        {
            arr[i] = sc.nextInt();
        }
        int sum = 0;
        for (int i = 0; i<n ;i++)
        {
            sum += arr[i];
        }
        float avg = (float) sum / n;
        System.out.println();
        System.out.println("Sum: " + sum);
```

```
        System.out.println("Average: " + avg);
    }
}
```

Output:



5. WAP to represent ArrayList class.

Code:

```
import java.util.ArrayList;
public class lab2_q5 {
public static void main(String[] args) {
ArrayList<String> names = new ArrayList<>();
names.add("Saumyaa");
names.add("Aaron");
names.add("Kanchan");
names.add("Yati");
names.add("Ananya");
names.add("Anam");
System.out.println("The names are: " + names);
System.out.println("The first name is: " + names.get(0));
System.out.println("The last name is: " + names.get(names.size() - 1));
names.set(1, "Kanchan");
System.out.println("The names after replacing the second name are: " + names);
```

```
names.remove(2);

System.out.println("The names after removing the third name are: " + names);

names.add(1, "Aaron");

System.out.println("The names after inserting a new name at index 1 are: " +
names);

}

}
```

Output:

Output Clear

```
java -cp /tmp/5AWI2VOF5L lab2_q5
The names are: [Saumyaa, Aaron, Kanchan, Yati, Ananya, Anam]
The first name is: Saumyaa
The last name is: Anam
The names after replacing the second name are: [Saumyaa, Kanchan,
    Kanchan, Yati, Ananya, Anam]The names after removing the third name
    are: [Saumyaa, Kanchan, Yati, Ananya, Anam]The names after
    inserting a new name at index 1 are: [Saumyaa, Aaron, Kanchan, Yati
    , Ananya, Anam]
```

LAB 3

Date:

6. WAP to check whether a string is palindrome or not.

Code:

```
import java.util.*;
public class Lab3_1
{
   static boolean Palindrome(String str)
   {
      String str1 = str.toLowerCase();
      int a = 0;
      int b = str1.length() - 1;
      while (a < b)
      {
         if (str1.charAt(a) != str1.charAt(b))
         {
            return false;
         }
         a++;
         b--;
      }
      return true;
   }
   public static void main(String[] args)
   {
      Scanner obj = new Scanner(System.in);
      System.out.print("Enter a string: ");
      String s = obj.nextLine();

      if (Palindrome(s))
```

```
        {
            System.out.println("palindrome string");
        }
        else
        {
            System.out.println("not a palindrome");
        }
    }
}
```

Output:

Output

```
java -cp /tmp/QRi1SIkPnf Lab3_1
Enter a string: kanchan
not a palindrome
```

Output

```
java -cp /tmp/QRi1SIkPnf Lab3_1
Enter a string: wowow
palindrome string
```

7. Write a java program for sorting a given list of names in ascending order.

Code:

```
import java.util.Scanner;
public class Lab3_2
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of names to be placed in the array: ");
        int n = sc.nextInt();
        sc.nextLine();
        String arr[] = new String[n];
        System.out.println("Enter the names: ");
        for (int i = 0; i < n; i++)
        {
            arr[i] = sc.nextLine();
        }
        for (int i = 0; i < n - 1; i++)
        {
            for (int j = 0; j < n - i - 1; j++)
            {
                if (arr[j].compareTo(arr[j + 1]) > 0)
                {
                    String temp = arr[j];
                    arr[j] = arr[j + 1];
```

```
                arr[j + 1] = temp;
            }
        }
    }
    System.out.println("Sorted names:");
    for (int a=0;a<n;a++)
    {
        System.out.println(arr[a]);
    }
  }
}
```

Output:

```
Output
java -cp /tmp/QRi1SIkPnf Lab3_2
Enter the number of names to be placed in the array: 3
Enter the names:
yati
kanchan
saumyaa
Sorted names:
kanchan
saumyaa
yati
```

LAB 4

Date:

8. Design three classes: Student, Exam and Result. The student class has data members such as roll no, name etc. Create a class Exam by inheriting the Student class. The Exam class adds data members representing the marks scored in six subjects. Derive the Result from class Exam and it has its own members such as total marks and average. Calculate the total marks and average.

Code:

```
import java.util.*;

class Student{

    int rno;

    String name;

}

class Exam extends Student{

    int
English_marks,Hindi_marks,SST_marks,Maths_marks,Science_marks,PEd_marks;

}

class Result extends Exam{

    int tmarks,avgmarks;

    void get_info(){

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter name of student: ");

        name=sc.next();

        System.out.print("Enter the roll no. of student: ");

        rno=sc.nextInt();

        System.out.print("Enter English marks: ");

        English_marks=sc.nextInt();

        System.out.print("Enter Hindi marks: ");

        Hindi_marks=sc.nextInt();
```

```
        System.out.print("Enter SST marks: ");
        SST_marks=sc.nextInt();
        System.out.print("Enter Maths marks: ");
        Maths_marks=sc.nextInt();
        System.out.print("Enter Science marks: ");
        Science_marks=sc.nextInt();
        System.out.print("Enter Physical Ed marks: ");
        PEd_marks=sc.nextInt();
        sc.close();

tmarks=English_marks+Hindi_marks+SST_marks+Maths_marks+Science_marks+P
Ed_marks;
        avgmarks=tmarks/6;
    }
    void tresult(){
        System.out.println("The total marks are: "+tmarks);
        System.out.println("The average marks are: "+avgmarks);
    }
}
public class Project{
public static void main(String[] args) {
    Result obj=new Result();
    obj.get_info();
    obj.tresult();
    }
}
```

Output:

```
import java.util.*;
class Student{
    int rno;
    String name;
}
class Exam extends Student{
    int
English_marks,Hindi_marks,SST_marks,Maths_marks,Science_marks,PEd_marks;
}
class Result extends Exam{
    int tmarks,avgmarks;
    void get_info(){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter name of student: ");
        name=sc.next();
        System.out.print("Enter the roll no. of student: ");
        rno=sc.nextInt();
        System.out.print("Enter English marks: ");
        English_marks=sc.nextInt();
        System.out.print("Enter Hindi marks: ");
        Hindi_marks=sc.nextInt();
        System.out.print("Enter SST marks: ");
        SST_marks=sc.nextInt();
        System.out.print("Enter Maths marks: ");
        Maths_marks=sc.nextInt();
        System.out.print("Enter Science marks: ");
```

```
        Science_marks=sc.nextInt();
        System.out.print("Enter Physical Ed marks: ");
        PEd_marks=sc.nextInt();
        sc.close();

tmarks=English_marks+Hindi_marks+SST_marks+Maths_marks+Science_marks+PEd_marks;
        avgmarks=tmarks/6;
    }
    void tresult(){
        System.out.println("The total marks are: "+tmarks);
        System.out.println("The average marks are: "+avgmarks);
    }
}
public class Project{
public static void main(String[] args) {
    Result obj=new Result();
    obj.get_info();
    obj.tresult();
    }
}
```

Output:

```
Enter the roll no. of student: 314
Enter English marks: 78
Enter Hindi marks: 76
Enter SST marks: 77
Enter Maths marks: 79
Enter Science marks: 76
Enter Physical Ed marks: 77
The total marks are: 463
The average marks are: 77
```

9. WAP to represent abstract class with example.

Code:

```
abstract class A{
    int x=5;
    abstract void show();
    void set(int y){
        x=y;
    }
}
class B extends A{
    void show(){
        System.out.println("New value of x is: "+x);
    }
}
public class Project2 {
    public static void main(String[] args) {
        B obj=new B();
        System.out.println("Current value of x is: "+obj.x);
        obj.set(10);
        obj.show();
    }
}
```

Output:

```
Current value of x is: 5
New value of x is: 10
```

10. Write a java program to create an abstract class named shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle,, Triangle and Circle such that each one of the classes extends the class shape. Each one of the class contains only the method print Area() that print the area of the given shape

Code:

```
import java.util.*;
abstract class Shape{
    int n,m;
    abstract void Area();
}
class Rectangle extends Shape{
    void get_rinfo(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter length of rectangle: ");
        n=sc.nextInt();
        System.out.println("Enter breadth of rectangle: ");
        m=sc.nextInt();
    }
    void Area(){
        System.out.println("The area of the rectangle is: "+n*m);
    }
}
class Triangle extends Shape{

    void get_tinfo(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter height of triangle: ");
        n=sc.nextInt();
        System.out.println("Enter base of triangle: ");
```

```
        m=sc.nextInt();
    }
    void Area(){
        System.out.println("The area of the triangle is: "+0.5*n*m);
    }
}
class Circle extends Shape{

    void get_cinfo(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter radius of circle: ");
        n=sc.nextInt();
    }
    void Area(){
        System.out.println("The area of the circle is: "+3.14*n*n);
    }
}
public class Project3 {
    public static void main(String[] args) {
        String c;
        Scanner sc=new Scanner(System.in);
        System.out.print("What shape?: r-rectangle, t-triangle, c-circle: ");
        c=sc.next();
        if(c.equals("r")){
            Rectangle r1=new Rectangle();
            r1.get_rinfo();
            r1.Area();
        }
        else if(c.equals("t")){
```

```
            Triangle t1= new Triangle();
            t1.get_tinfo();
            t1.Area();
        }
        else if(c.equals("c")){
            Circle c1=new Circle();
            c1.get_cinfo();
            c1.Area();
        }
        else{
            System.out.println("Invalid choice");
        }
    }
}
```

Output:

```
What shape?: r-rectangle, t-triangle, c-circle: r
Enter length of rectangle:
5
Enter breadth of rectangle:
4
The area of the rectangle is: 20

What shape?: r-rectangle, t-triangle, c-circle: t
Enter height of triangle:
5
Enter base of triangle:
4
The area of the triangle is: 10.0
What shape?: r-rectangle, t-triangle, c-circle: c
Enter radius of circle:
5
The area of the circle is: 78.5
```

LAB 5

Date:

1. Calculate area of different figures like circle, rectangle, square, triangle using function overloading.

Code:

```
import java.util.*;
class Test{
    static void Area(float x,float y, int ch){
        if(ch==2){
            System.out.println("Area of Rectangle: "+(x*y));
        }
        else if(ch==4){
            System.out.println("Area of Triangle: "+(0.5 * x*y));
        }
    }
    static void Area(float x,int ch){
        if(ch==1)
        {
            System.out.println("Area of Square: "+(x*x));
        }
        else if(ch==3)
        {
            System.out.println("Area of Circle: "+(3.14 * x*x));
        }
    }
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("1.Square\n 2.Rectangle\n 3.Circle\n 4.Triangle");
        System.out.print("Enter your choice: ");
        int ch = sc.nextInt();
        float x=0, y=0;
```

```
if(ch == 1){
    System.out.print("Enter side: ");
    x = sc.nextFloat();
    System.out.println();
    Area(x,ch);
}
else if(ch == 2){
    System.out.print("Enter length: ");
    x = sc.nextFloat();
    System.out.println();
    System.out.print("Enter breadth: ");
    y = sc.nextFloat();
    System.out.println();
    Area(x,y,ch);
}
else if(ch == 3){
    System.out.print("Enter radius: ");
    x = sc.nextFloat();
    System.out.println();
    Area(x,ch);
}
else if(ch ==4){
    System.out.print("Enter base: ");
    x = sc.nextFloat();
    System.out.println();
    System.out.print("Enter height: ");
    y = sc.nextFloat();
    System.out.println();
    Area(x,y,ch);
}
```

```
        else{
            System.out.println("Invalid choice");
        }
    }
}
```

Output:

```
Output
java -cp /tmp/HSnLgjzy66 Test
1.Square
 2.Rectangle
 3.Circle
 4.Triangle
Enter your choice: 1
Enter side: 3.4

Area of Square: 11.56
```

```
1.Square
 2.Rectangle
 3.Circle
 4.Triangle
Enter your choice: 4
Enter base: 3

Enter height: 5

Area of Triangle: 7.5
```

Lab 6

Date:

1. Write a program to implement multiple inheritance using interfaces.

Code:

```
import java.util.*;
interface A
{
    int a=10;
    void printA();
}
interface B
{
    int b=20;
    void printB();
}
class C implements A,B
{

    public void printA()
    {
        System.out.println("Value of a: "+a);
    }
    public void printB()
    {
        System.out.println("Value of b: "+b);
    }
}
class JavaApplication22
{
```

```
    public static void main(String[] args)
    {
        C obj = new C();
        obj.printA();
        obj.printB();
    }

}
```

Output:

```
Output

java -cp /tmp/5mgkDywLqI JavaApplication22
Value of a: 10
Value of b: 20
```

Lab 7

Date:

1. Write a program to implement exception handling.

```
public class JavaApplication20
{
    public static void main(String[] args)
    {
        try
        {
            int arr[]={1,2,3,4};
            int y = arr[3];
            int a = 5/0;
        }
        catch(ArithmeticException e)
        {
            e.printStackTrace();
            System.out.println("Arithmetic Error (division by zero)");
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            e.printStackTrace();
            System.out.println("AOBE Error");
        }
    }
}
```

Output:

```
Output
java -cp /tmp/CREakOz33a JavaApplication20
java.lang.ArithmeticException: / by zero
    at JavaApplication20.main(JavaApplication20.java:9)
Arithmetic Error (division by zero)
```

2. Write a program to handle your own exception

Code:

```
class OwnExcep extends Exception
{

}

public class JavaApplication21
{
    public static void main(String[] args)
    {
        try
        {
            int age = 17;
            if(age<18)
            {
                throw new OwnExcep();
            }
        }
        catch(OwnExcep e)    //own exception
        //catch(Exception e)   //general exception
        {
                System.out.print("Age is less than 18");
        }
    }
}
```

Output:

```
Output
java -cp /tmp/CREakOz33a JavaApplication21
Age is less than 18
```

LAB 8

Date:

1. Write a java program for producer and consumer problem using Threads.

Code:

```
import java.util.*;
class Queue {
private LinkedList<Integer> list = new LinkedList<>();
int capacity;
public Queue(int capacity) {
this.capacity = capacity;
}
public synchronized void put(int value) throws InterruptedException {
while (list.size() == capacity)
wait();
list.add(value);
System.out.println("Producer has produced: " + value);
notify();
}
public synchronized int get() throws InterruptedException {
while (list.size() == 0) {
wait();
}
int value = list.removeFirst();
System.out.println("Consumer has consumed: " + value);
notify();
return value;
}
}
class Producer extends Thread {
private Queue queue;
public Producer(Queue queue) {
this.queue = queue;
}
public void run() {
try {
for (int i = 0; i < queue.capacity; i++) {
queue.put(i + 1);
Thread.sleep(1000);
}
} catch (InterruptedException e) {
e.printStackTrace();
}
}
}
class Consumer extends Thread {
private Queue queue;
```

```
  public Consumer(Queue queue) {
  this.queue = queue;
  }
  public void run() {
  try {
  for (int i = 0; i < queue.capacity; i++) {
  queue.get();
  Thread.sleep(1000);
  }
  } catch (InterruptedException e) {
  e.printStackTrace();
  }
  }
  }
  public class ProducerConsumer {
  public static void main(String[] args) {
  Queue queue = new Queue(5);
  Producer producer = new Producer(queue);
  Consumer consumer = new Consumer(queue);
  Thread producerThread = new Thread(producer);
  Thread consumerThread = new Thread(consumer);
  producerThread.start();
  consumerThread.start();
  }
}
```

Output:

```
Producer has produced: 1
Consumer has consumed: 1
Producer has produced: 2
Consumer has consumed: 2
Producer has produced: 3
Consumer has consumed: 3
Producer has produced: 4
Consumer has consumed: 4
Producer has produced: 5
Consumer has consumed: 5
```

2. Write a Java program that implements a multi-thread application that has three threads.

Code:

```
class demo extends Thread{
private int count=5;
Thread t;
demo(String s,int p){
t=new Thread(this,s);
t.setPriority(p);
t.start();
}
public void run(){
while(true){
System.out.println("#"+t.getName()+":"+count);
if(--count==0) return;
}
}
}
public class Test{
public static void main(String[] args) {
new demo("T1",Thread.NORM_PRIORITY);
new demo("T2",Thread.MAX_PRIORITY);
new demo("T3",Thread.MIN_PRIORITY);
}
}
```

Output:

```
#T2:5
#T2:4
#T1:5
#T3:5
#T3:4
#T2:3
#T1:4
#T1:3
#T3:3
#T2:2
#T1:2
#T3:2
#T2:1
#T1:1
#T3:1
```

3. Write a Java program that implements a multi-threaded program has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd the third thread will print the value of cube of the number.

Code:

```
import java.util.Random;
class Square extends Thread
{
int x;
Square(int n)
{
x = n;
}
public void run()
{
int sqr = x * x;
System.out.println("Square of " + x + " = " + sqr );
}
}
class Cube extends Thread
{
int x;
Cube(int n)
{
x = n;
}
public void run()
{
int cub = x * x * x;
System.out.println("Cube of " + x + " = " + cub );
}
}
class Number extends Thread
{
public void run()
{
Random random = new Random();
for(int i =0; i<10; i++)
{
int rInt = random.nextInt(100);
System.out.println("Random Integer generated : " + rInt);
if(rInt%2==0){
Square s = new Square(rInt);
s.start();
}
else{
Cube c = new Cube(rInt);
c.start();
```

```
        }
        try {
        Thread.sleep(1000);
        } catch (InterruptedException ex) {
        System.out.println(ex);
        }
        }
        }
        }
        public class Test {
        public static void main(String args[])
        {
        Number n = new Number();
        n.start();
        }
}
```

Output:

```
Random Integer generated : 39
Cube of 39 = 59319
Random Integer generated : 7
Cube of 7 = 343
Random Integer generated : 61
Cube of 61 = 226981
Random Integer generated : 88
Square of 88 = 7744
Random Integer generated : 63
Cube of 63 = 250047
Random Integer generated : 3
Cube of 3 = 27
Random Integer generated : 56
Square of 56 = 3136
Random Integer generated : 90
Square of 90 = 8100
Random Integer generated : 92
Square of 92 = 8464
Random Integer generated : 55
Cube of 55 = 166375
```

LAB 9

Date:

1. Write a java program that loads data from a text file using hash table.

Code:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Hashtable;
public class Test {
public static void main(String[] args) {
Hashtable<String, String> phoneBook = new Hashtable<>();
String filePath = "phonebook.txt";
try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
String line;
while ((line = br.readLine()) != null) {
String[] parts = line.split("\\s+", 2);
if (parts.length >= 2) {
String phoneNumber = parts[0];
String name = parts[1];
phoneBook.put(phoneNumber, name);
}
}
} catch (IOException e) {
System.err.println("Error reading file: " + e.getMessage());
}
System.out.println("Phone Book:");
for (String phoneNumber : phoneBook.keySet()) {
String name = phoneBook.get(phoneNumber);
System.out.println("Phone Number: " + phoneNumber + ", Name: " + name);
}
}
}
```

Code:

```
Phone Book:
Phone Number: 9876543210, Name: Annabeth Chase
Phone Number: 2223334445, Name: Will Solace
Phone Number: 1234567890, Name: Percy Jackson
Phone Number: 1112223334, Name: Piper Mclean
Phone Number: 5551234567, Name: Nico Di Angelo
Phone Number: 9998887776, Name: Thalia Grace
Phone Number: 4443332225, Name: Jason Grace
Phone Number: 6665554443, Name: Hazel Velezque
Phone Number: 8887776661, Name: Frank Zhang
Phone Number: 7778889992, Name: Leo Valdez
```

2. Write a program to create a file in Java using FileOutputStream class and enter byte data into it.

Code:

```
import java.io.FileOutputStream;
import java.io.IOException;
public class Test
{
public static void main(String args[])
{
try
{
FileOutputStream f=new FileOutputStream("hellojava.txt");
f.write(69);
f.close();
System.out.println("Data written in the file ");
}
catch(IOException e)
{
System.out.println(e);
}
}
}
```

Output:

```
Data written in the file
```

3. Write a program to create a java file using FileOutputStream class and enter a string into it.

Code:

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Scanner;
public class Test {
public static void main(String[] args)
{
try
{
String data;
Scanner s= new Scanner(System.in);
System.out.print("Enter the string:");
data=s.nextLine();
try (FileOutputStream x = new FileOutputStream("stringdata.txt")) {
byte arr[]=data.getBytes();
x.write(arr);
System.out.println("Data has been succesfully written onto the file");
```

```
  }
  }
  catch(IOException e)
  {
  System.out.println(e);
  }
  }
}
```

Output:

```
Enter the string:Hello
Data has been succesfully written onto the file
```

4. Open a file using FileInputStream read its content and display on screen.

Code:

```
  import java.io.FileInputStream;
  import java.io.FileOutputStream;
  import java.io.IOException;
  import java.util.*;
  public class Test{
  public static void main(String[] args) {
  try{
  Scanner sc= new Scanner(System.in);
  System.out.print("Enter the string you want to write to the file: ");
  String a=sc.next();
  FileOutputStream fo= new FileOutputStream("newfile.txt");
  byte arr[]=a.getBytes();
  fo.write(arr);
  fo.close();
  FileInputStream fi=new FileInputStream("newfile.txt");
  int ch;
  System.out.println("-----------------------------------------------------");
  System.out.print("The data you have entered into the file was: ");
  while ((ch = fi.read()) != -1) {
  System.out.print((char) ch);
  }
  }
  catch (IOException e) {
  System.out.println("An error occurred: " + e.getMessage());
  }
  }
}
```

Output:

```
Enter the string you want to write to the file: AmityUniversity
--------------------------------------------------
The data you have entered into the file was: AmityUniversity
```