RAJAGIRI COLLEGE OF SOCIAL SCIENCES

(AUTONOMOUS) , KALAMASSERY

SEMESTER 2

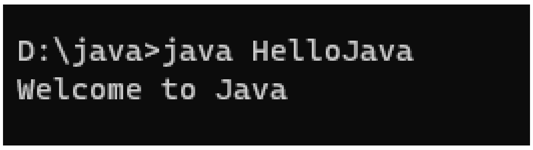
JAVA RECORD

K DEVI KRISHNA

MSc CS (DA)

1.Write a program to print ‘Welcome to Java’.

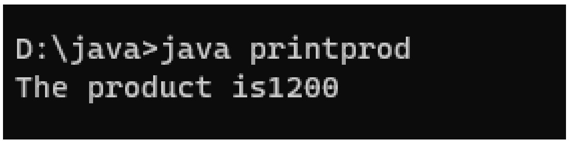
```
class HelloJava {
    public static void main(String[] args) {
        System.out.println("Welcome to Java");
    }
}
```



```
D:\java>java HelloJava
Welcome to Java
```

2. Write a program to display two numbers received as command line argument, and print its product.

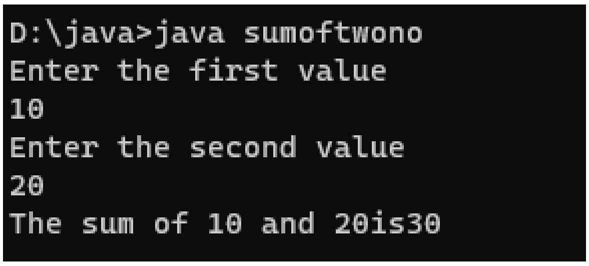
```
class printprod
 {
    public static void main(String[] args)
 {
  int i=30;
  int j=40;
 int prod=i*j;
System.out.println("The product is"+prod);
    }
}
```



```
D:\java>java printprod
The product is1200
```

3. Write a program to read two numbers and display the output in the form of ‘Sum of 2 and 3 is 5.

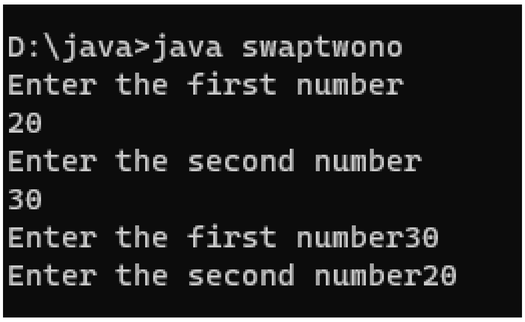
```
import java.io.*;
public class sumoftwono
{
public static void main(String args[])
{
try
{
String s;
int sum;
int x,y;
DataInputStream dis=new DataInputStream(System.in);
System.out.println("Enter the first value");
s=dis.readLine();
x=Integer.parseInt(s);
System.out.println("Enter the second value");
s=dis.readLine();
y=Integer.parseInt(s);
sum=x+y;
System.out.println("The sum of "+x+"and  "+y+"is"+sum);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```



```
D:\java>java sumoftwono
Enter the first value
10
Enter the second value
20
The sum of 10 and 20is30
```

4. Write a program to accept two numbers from the keyboard and swap them.

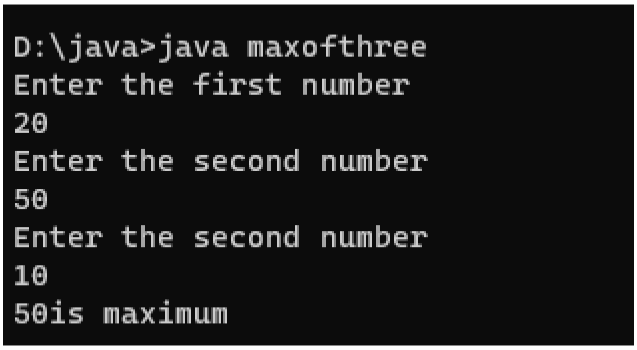
```
import java.io.*;
public class swaptwono
{
public static void main(String args[])
{
try
{
String s;
int x,y;
int temp;
System.out.println("Enter the first number");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
x=Integer.parseInt(s);
System.out.println("Enter the second number");
s=dis.readLine();
y=Integer.parseInt(s);
temp=x;
x=y;
y=temp;
System.out.println("Enter the first number"+x);
System.out.println("Enter the second number"+y);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```



```
D:\java>java swaptwono
Enter the first number
20
Enter the second number
30
Enter the first number30
Enter the second number20
```

5. Write a program to read three numbers and the maximum.

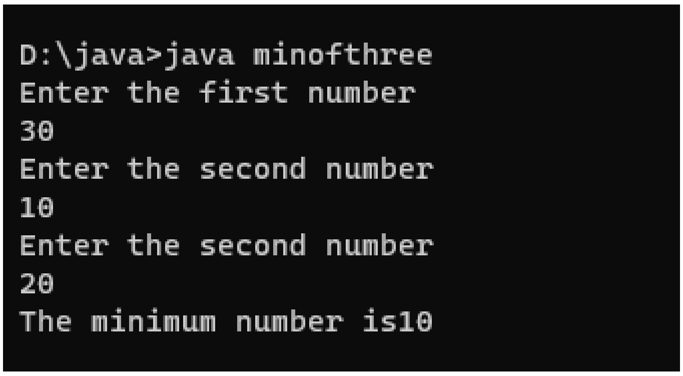
```
import java.io.*;
public class maxofthree
{
public static void main(String args[])
{
try
{
String s;
int x,y,z;
System.out.println("Enter the first number");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
x=Integer.parseInt(s);
System.out.println("Enter the second number");
s=dis.readLine();
y=Integer.parseInt(s);
System.out.println("Enter the second number");
s=dis.readLine();
z=Integer.parseInt(s);
if(x>=y&&x>=z)
System.out.println(x+"is maximum");
else if(y>=x&&y>=z)
System.out.println(y+"is maximum");
else
System.out.println(z+"is maximum");
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```



```
D:\java>java maxofthree
Enter the first number
20
Enter the second number
50
Enter the second number
10
50is maximum
```

6. Find the minimum of three numbers using a single statement.

```
import java.io.*;
public class minofthree
{
public static void main(String args[])
{
try
{
String s;
int x,y,z;
System.out.println("Enter the first number");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
x=Integer.parseInt(s);
System.out.println("Enter the second number");
s=dis.readLine();
y=Integer.parseInt(s);
System.out.println("Enter the second number");
s=dis.readLine();
z=Integer.parseInt(s);
int temp=x<y?x:y;
int min=z<temp?z:temp;
System.out.println("The minimum number is"+min);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

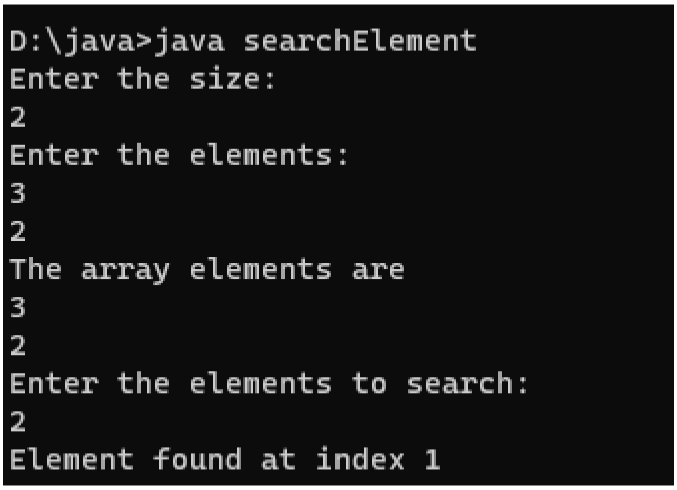


```
D:\java>java minofthree
Enter the first number
30
Enter the second number
10
Enter the second number
20
The minimum number is10
```

7. Write a program to search for a given element in an array.

```
import java.io.*;
class searchElement
 {
   public static void main(String[] args)
 {
 try
 {
String s;
int i;
int a[];
int found=0;
System.out.println("Enter the size:");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
int n=Integer.parseInt(s);
a=new int[n];

System.out.println("Enter the elements:");
for(i=0;i<n;i++)
{
s=dis.readLine();
a[i]=Integer.parseInt(s);
}
System.out.println("The array elements are");
for(i=0;i<n;i++)
{
System.out.println(a[i]);
}
System.out.println("Enter the elements to search:");
s=dis.readLine();
int search=Integer.parseInt(s);
for (i = 0; i <n; i++) {
            if (a[i] == search) {
               found = 1;
               System.out.println("Element found at index " + i);
               break;
            }
```

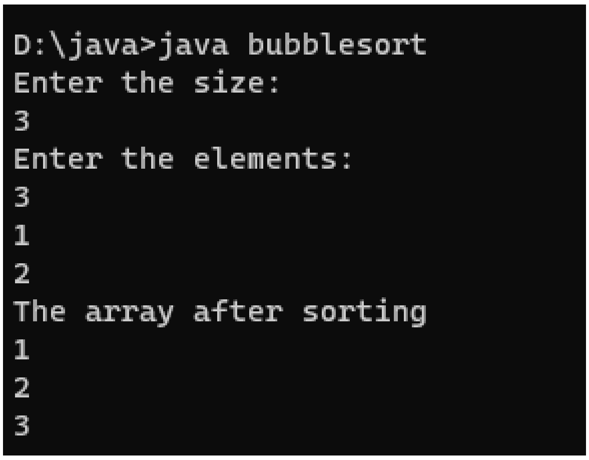
```
        }

        if (found==0) {
            System.out.println("Element not found in the array");
        }
}
catch(Exception e)
{
System.out.println(e);
}
}}
```



```
D:\java>java searchElement
Enter the size:
2
Enter the elements:
3
2
The array elements are
3
2
Enter the elements to search:
2
Element found at index 1
```

8. Write a program to sort elements in an array in ascending order.

```
import java.io.*;
class bubbleSort
 {
    public static void main(String[] args)
 {
 try
 {
String s;
int i,j;
int temp;
int a[];
System.out.println("Enter the size:");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
int n=Integer.parseInt(s);
a=new int[n];
System.out.println("Enter the elements:");
for(i=0;i<n;i++)
{
s=dis.readLine();
a[i]=Integer.parseInt(s);
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
if(a[i]<a[j])
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
System.out.println("The array after sorting");
for(i=0;i<n;i++)
{
System.out.println(a[i]);
}
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```



```
D:\java>java bubblesort
Enter the size:
3
Enter the elements:
3
1
2
The array after sorting
1
2
3
```

9. Write a program to print the row wise and column wise sum of a 2D array.

1 2 3 | 6

2 1 1 |4

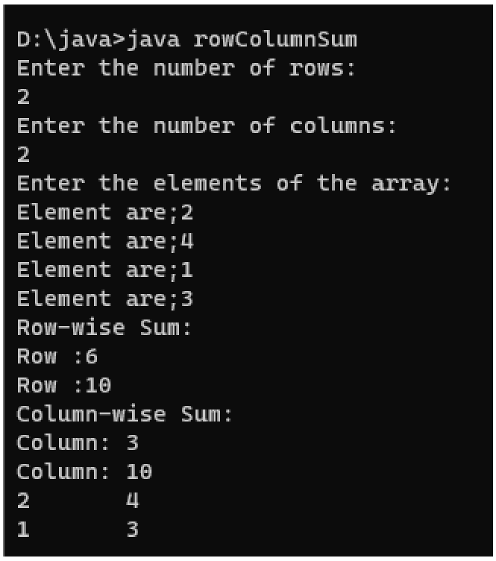
. . .

3 3 4 |10

```
import java.io.*;
class rowColumnSum
 {
   public static void main(String[] args)
 {
```

```
 try
 {
String s;
int i,j;
System.out.println("Enter the number of rows:");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
int row=Integer.parseInt(s);
System.out.println("Enter the number of columns:");
s=dis.readLine();
int col=Integer.parseInt(s);
int a[][]= new int[row][col];
System.out.println("Enter the elements of the array:");
 for (i = 0; i < row; i++) {
  for (j = 0; j < col; j++) {
    System.out.print("Element are;");
s=dis.readLine();
 a[i][j] = Integer.parseInt(s);
}
        }
int rowSum = 0;
System.out.println("Row-wise Sum:");
 for (i=0;i<row;i++) {
 for (j=0;j<col; j++) {
rowSum=rowSum+a[i][j];
            }
            System.out.println("Row :" + rowSum);
        }
 int colSum = 0;
System.out.println("Column-wise Sum:");
    for (j=0;j<col; j++) {
      for (i=0;i<row;i++) {
                colSum=colSum+a[i][j];
            }
 System.out.println("Column: " + colSum);
        }

 for (i=0;i<row;i++) {
            for(j=0;j<col;j++) {
                System.out.print(a[i][j] + "\t");
            }
            System.out.println();
        }
   }
catch(Exception e)
{
System.out.println(e);
}
```

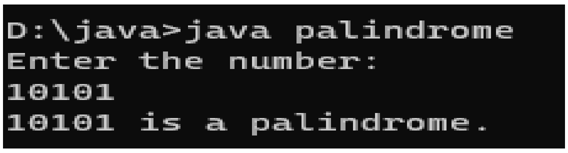
```
}
}
```



```
D:\java>java rowColumnSum
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of the array:
Element are;2
Element are;4
Element are;1
Element are;3
Row-wise Sum:
Row :6
Row :10
Column-wise Sum:
Column: 3
Column: 10
2       4
1       3
```

10. Write a program with two functions to check for an integer palindrome. (Function1 should reverse the integer. Function2 should return 1,if it is a palindrome or else 0.)

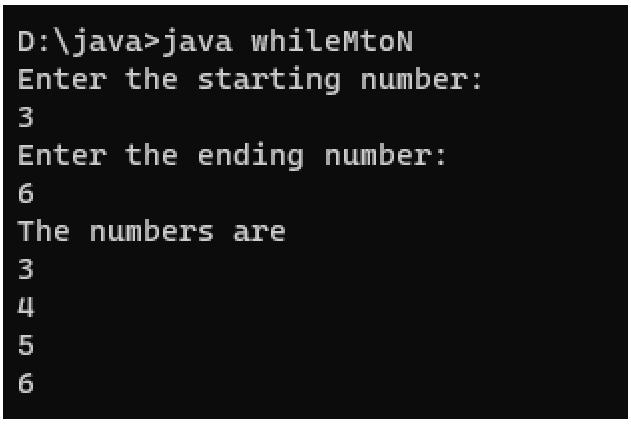
```
import java.io.*;
public class palindrome
{
public static int reverse(int num)
{
int rev=0;
while(num>0)
{
int x=num%10;
rev=rev*10+x;
num=num/10;
}
return rev;
}
public static void main(String[] args)
 {
 try
 {
String s;
int num;
DataInputStream dis=new DataInputStream(System.in);
System.out.println("Enter the number:");
s=dis.readLine();
num=Integer.parseInt(s);
int rev=reverse(num);
int result = (rev == num) ? 1 : 0;
if (result == 1) {
        System.out.println(num + " is a palindrome.");
        } else {
            System.out.println(num + " is not a palindrome.");
        }
}
catch(Exception e)
{
System.out.println(e);
}}
}
```



```
D:\java>java palindrome
Enter the number:
10101
10101 is a palindrome.
```

11. Write a program to display numbers from m to n using single while loop.

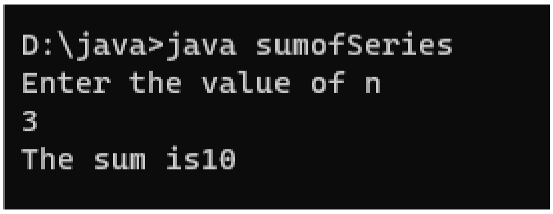
```
import java.io.*;
public class whileMtoN
{
public static void main(String[] args)
 {
 try
 {
String s;
int x,y;
DataInputStream dis=new DataInputStream(System.in);
System.out.println("Enter the starting number:");
s=dis.readLine();
x=Integer.parseInt(s);
System.out.println("Enter the ending number:");
s=dis.readLine();
y=Integer.parseInt(s);
System.out.println("The numbers are");
while(x<=y)
{
System.out.println(x);
x++;
}
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```



```
D:\java>java whileMtoN
Enter the starting number:
3
Enter the ending number:
6
The numbers are
3
4
5
6
```

12. Write a program to find the sum of the series 1+(1+2)+(1+2+3)+........... +(1+2+3+…+n) using a single while
 loop.

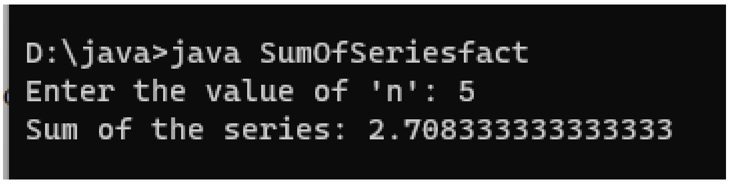
```
import java.io.*;
public class sumofSeries
{
public static void main(String args[])
{
try
{
String s;
int n,i=0,j;
int sum=0;
int num=0;
DataInputStream dis=new DataInputStream(System.in);
System.out.println("Enter the value of n");
s=dis.readLine();
n=Integer.parseInt(s);
while(i<=n)
{
num=num+i;
sum=sum+num;
i++;
}
System.out.println("The sum is"+sum);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```



```
D:\java>java sumofSeries
Enter the value of n
3
The sum is10
```

13. Write a program to find the sum of 1+2/2!+3/3!+4/4!++n/n! using a single for loop.

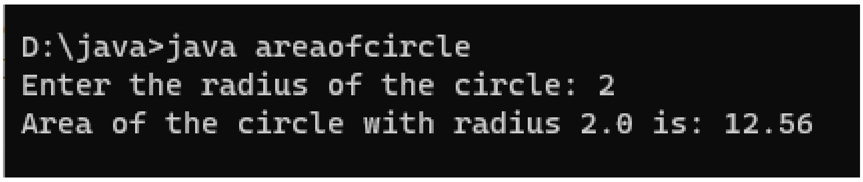
```
import java.util.Scanner;

public class SumOfSeriesfact {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the value of 'n': ");
        int n = scanner.nextInt();
        double sum = 1.0;
            for (int i = 2, factorial = 1; i <= n; i++) {
            factorial *= i;
            sum += (double) i / factorial;
        }
       System.out.println("Sum of the series: " + sum);
    }
}
```



```
D:\java>java SumOfSeriesfact
Enter the value of 'n': 5
Sum of the series: 2.708333333333333
```

14. Write a program to calculate area of a circle (functions with no argument and no return type.)

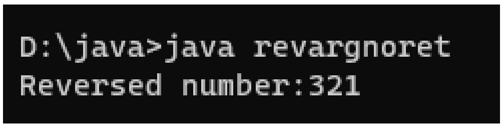
```
import java.util.*;
class area {
public void calculateArea() {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the radius of the circle: ");
double radius = scanner.nextDouble();
double area = 3.14 * radius * radius;
System.out.println("Area of the circle with radius " + radius + " is: " + area);
}
}

public class areaofcircle {
public static void main(String[] args) {
area a = new area();
a.calculateArea();
}
}
```



```
D:\java>java areaofcircle
Enter the radius of the circle: 2
Area of the circle with radius 2.0 is: 12.56
```

15. Write a program to reverse a number (functions with argument and no return type.)

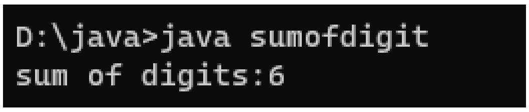
```
class rever
{
public void revers(int x)
{
int reverse=0;
while(x!=0)
{
int rev=x%10;
reverse=reverse*10+rev;
x/=10;
}
System.out.println("Reversed number:"+reverse);
}
}

public class revargnoret
{
public static void main(String args[])
{
rever r=new rever();
int a =123;
r.revers(a);
}
}
```



```
D:\java>java revargnoret
Reversed number:321
```

16. WAP to calculate sum of digits of a number (functions with argument and return type.)

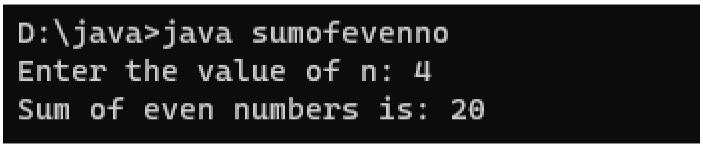
```
class test
{
public int digits(int x)
{
int sum=0;
while(x!=0)
{
int digit=x%10;
sum=sum+digit;
x/=10;
}
return sum;
}
}


public class sumofdigit
{
public static void main(String args[])
{
test t=new test();
int a =123;
int result=t.digits(a);
System.out.println("sum of digits:"+result);
}
}
```



```
D:\java>java sumofdigit
sum of digits:6
```

17. Write a program to calculate sum of n even numbers (functions with no argument and return type.)

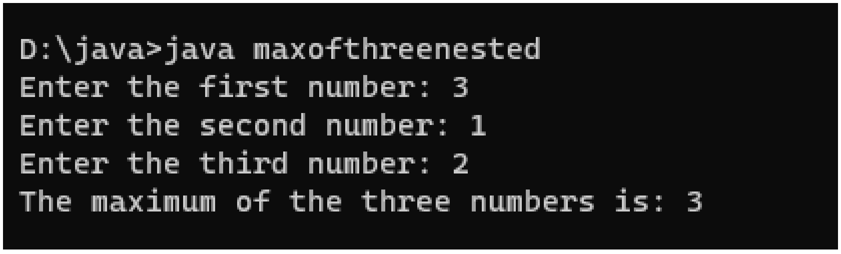
```
import java.io.*;

class test {
   public int calculateSum() throws IOException {
      BufferedReader reader = new BufferedReader(newInputStreamReader(System.in));
 System.out.print("Enter the value of n: ");
      int n = Integer.parseInt(reader.readLine());
      int sum = 0;
      for (int i = 1; i <= n; i++) {
         sum += 2 * i;
      }
      return sum;
   }
}

public class sumofevenno {
   public static void main(String args[]) throws IOException {
      test t = new test();
      int result = t.calculateSum();
      System.out.println("Sum of even numbers is: " + result);
   }
}
```



```
D:\java>java sumofevenno
Enter the value of n: 4
Sum of even numbers is: 20
```

18.Write a program with nested functions to find the maximum of three numbers. Function1 should take in two arguments and find the maximum. Function2 should take in the third number and the maximum from function1 to find the maximum.)

```
import java.io.*;

class MaxFinder {
   public int findMax(int a, int b) {
      return (a > b) ? a : b;
   }

   public int findMaxOfThree(int a,int b,int c) {
      int maxAB = findMax(a,b);
      return (c > maxAB) ? c : maxAB;
   }
}

public class maxofthreenested {
public static void main(String[] args) throws IOException {
 MaxFinder maxFinder = new MaxFinder();
 BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
      System.out.print("Enter the first number: ");
      int x= Integer.parseInt(reader.readLine());
      System.out.print("Enter the second number: ");
      int y = Integer.parseInt(reader.readLine());
      System.out.print("Enter the third number: ");
      int z = Integer.parseInt(reader.readLine());
      int result = maxFinder.findMaxOfThree(x,y,z);
      System.out.println("The maximum of the three numbers is: " + result);
   }
}
```

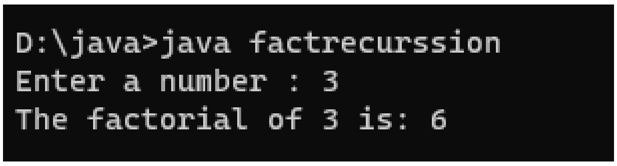


```
D:\java>java maxofthreenested
Enter the first number: 3
Enter the second number: 1
Enter the third number: 2
The maximum of the three numbers is: 3
```

19.Write a program to find the factorial of n, using recursion.

```
import java.util.*;
```

```
public class factrecurssion {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number : ");
        int n = scanner.nextInt();
        int result = fact(n);
        System.out.println("The factorial of " + n + " is: " + result);
    }
    public static int fact(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            return n * fact(n - 1);
        }
    }
}
```

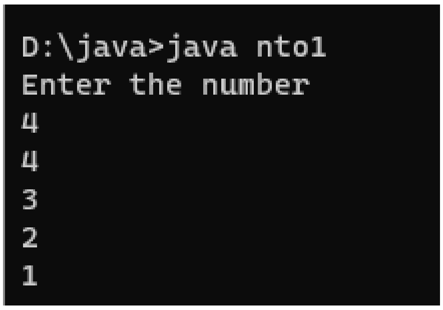


```
D:\java>java factrecurssion
Enter a number : 3
The factorial of 3 is: 6
```

20. Write a program to display numbers from n to 1 and vice versa, using recursion.

```
import java.io.*;
public class nto1
{
public static void disp(int n)
{
if(n>=1)
{
System.out.println(n);
disp(n-1);
}
}
public static void main(String args[])
{
try
{
String s;
int n;
DataInputStream dis=new DataInputStream(System.in);
System.out.println("Enter the number");
s=dis.readLine();
n=Integer.parseInt(s);
disp(n);
}
catch(Exception e)
{
```

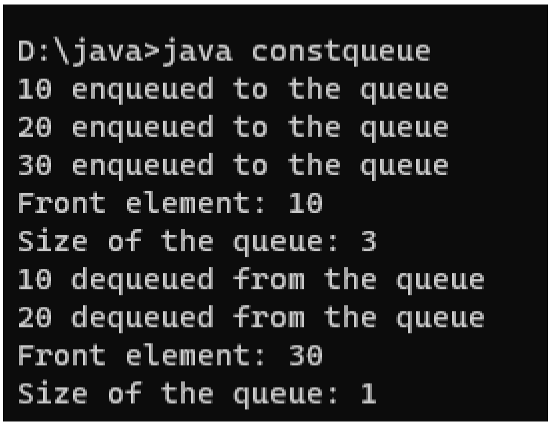
```
System.out.println(e);
}
}
}
```



```
D:\java>java nto1
Enter the number
4
4
3
2
1
```

21. Using constructors, implement the operations of a queue.

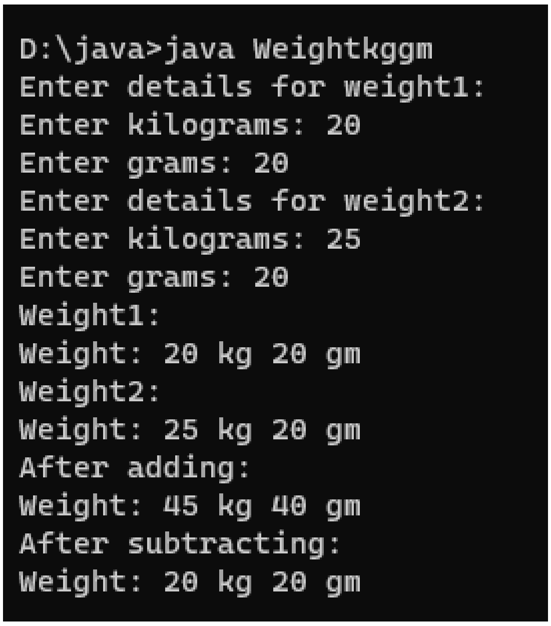
```
class Queue {
    private static final int MAX_SIZE = 100;
    private int front, rear, size;
    private int[] array;
    public Queue() {
        this.front = this.size = 0;
        this.rear = -1;
        this.array = new int[MAX_SIZE];
    }
    public boolean isEmpty() {
        return size == 0;
    }
    public boolean isFull() {
        return size == MAX_SIZE;
    }
    public int size() {
        return size;
    }
    public void enqueue(int item) {
        if (isFull()) {
            System.out.println("Queue is full. Cannot enqueue.");
            return;
        }
        rear = (rear + 1) % MAX_SIZE;
        array[rear] = item;
        size++;
        System.out.println(item + " enqueued to the queue");
    }
    public int dequeue() {
        if (isEmpty()) {
            System.out.println("Queue is empty. Cannot dequeue.");
            return -1; // Indicates an empty queue
        }
        int item = array[front];
        front = (front + 1) % MAX_SIZE;
        size--;
        return item;
    }


    public int peek() {
        if (isEmpty()) {
            System.out.println("Queue is empty. Cannot peek.");
            return -1; // Indicates an empty queue
        }

        return array[front];
    }
}

public class constqueue {
    public static void main(String[] args) {
        Queue queue = new Queue();
        queue.enqueue(10);
        queue.enqueue(20);
        queue.enqueue(30);
        System.out.println("Front element: " + queue.peek());
        System.out.println("Size of the queue: " + queue.size());
        System.out.println(queue.dequeue() + " dequeued from the queue");
        System.out.println(queue.dequeue() + " dequeued from the queue");
        System.out.println("Front element: " + queue.peek());
        System.out.println("Size of the queue: " + queue.size());
    }
}
```



```
D:\java>java constqueue
10 enqueued to the queue
20 enqueued to the queue
30 enqueued to the queue
Front element: 10
Size of the queue: 3
10 dequeued from the queue
20 dequeued from the queue
Front element: 30
Size of the queue: 1
```

22.Create a class “weight” having a kg and gm part. Provide functions for read, display, add and subtracting two weights.

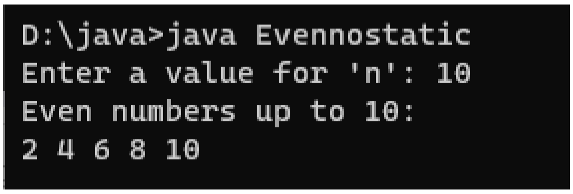
```
import java.util.Scanner;
class Weight {
    private int kg;
    private int gm;
    public void readWeight() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter kilograms: ");
        kg = scanner.nextInt();
        System.out.print("Enter grams: ");
        gm = scanner.nextInt();
    }

    public void displayWeight() {
        System.out.println("Weight: " + kg + " kg " + gm + " gm");
    }

    public void addWeight(Weight other) {
        kg += other.kg;
        gm += other.gm;
        if (gm >= 1000) {
            kg += gm / 1000;
            gm %= 1000;
        }
    }

    public void subtractWeight(Weight other) {
        kg -= other.kg;
        gm -= other.gm;
        if (gm < 0) {
            kg -= Math.abs(gm) / 1000 + 1;
            gm = 1000 - Math.abs(gm) % 1000;
        }
    }
}

public class Weightkggm {
    public static void main(String[] args) {
        Weight weight1 = new Weight();
        Weight weight2 = new Weight();
        System.out.println("Enter details for weight1:");
        weight1.readWeight();
        System.out.println("Enter details for weight2:");
        weight2.readWeight();
         System.out.println("Weight1:");
        weight1.displayWeight();
        System.out.println("Weight2:");
        weight2.displayWeight();
        weight1.addWeight(weight2);
        System.out.println("After adding:");
        weight1.displayWeight();
        weight1.subtractWeight(weight2);
        System.out.println("After subtracting:");
        weight1.displayWeight();
    }
}
```



```
D:\java>java Weightkggm
Enter details for weight1:
Enter kilograms: 20
Enter grams: 20
Enter details for weight2:
Enter kilograms: 25
Enter grams: 20
Weight1:
Weight: 20 kg 20 gm
Weight2:
Weight: 25 kg 20 gm
After adding:
Weight: 45 kg 40 gm
After subtracting:
Weight: 20 kg 20 gm
```

23. Write a program to display even numbers upto ‘n’ using a static function.

```
import java.util.Scanner;

public class Evennostatic {
    public static void displayEvenNumbers(int n) {
        System.out.println("Even numbers up to " + n + ":");
        for (int i = 2; i <= n; i += 2) {
            System.out.print(i + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a value for 'n': ");
        int n = scanner.nextInt();
       displayEvenNumbers(n);
    }
}
```

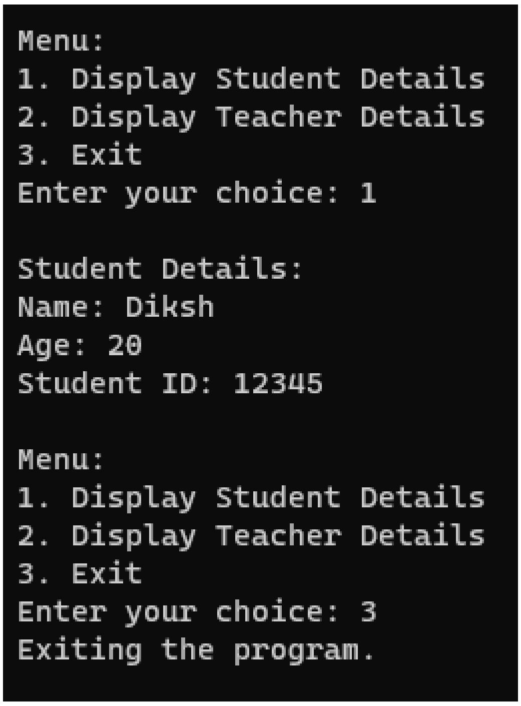


```
D:\java>java Evennostatic
Enter a value for 'n': 10
Even numbers up to 10:
2 4 6 8 10
```

24. Write a program (menu driven) to demonstrate method overriding in java, by displaying details of a student, and a teacher.

```
import java.util.Scanner;
class Person {
    protected String name;
    protected int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
class Student extends Person {
    private int studentId;
    public Student(String name, int age, int studentId) {
        super(name, age);
        this.studentId = studentId;
    }
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Student ID: " + studentId);
    }
}
class Teacher extends Person {
    private String subject;
    public Teacher(String name, int age, String subject) {
        super(name, age);
        this.subject = subject;
    }
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Subject: " + subject);
    }
}
public class methodoverriding {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;
        do {
            System.out.println("\nMenu:");
            System.out.println("1. Display Student Details");
            System.out.println("2. Display Teacher Details");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
```

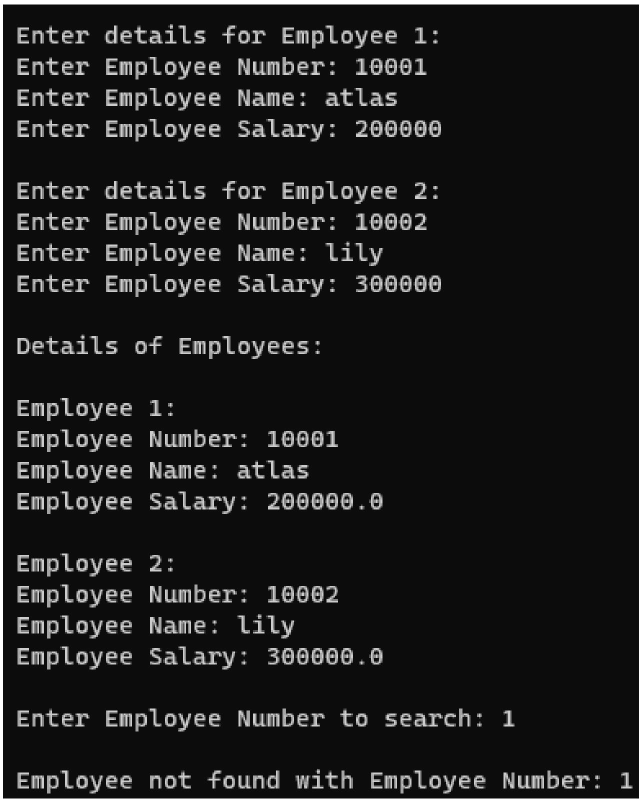
```
                Student student = new Student("Diksh", 20, 12345);
                System.out.println("\nStudent Details:");
                student.displayDetails();
                break;
              case 2:
              Teacher teacher = new Teacher("Mr.Dev", 35, "CS");
                System.out.println("\nTeacher Details:");
                teacher.displayDetails();
                break;
              case 3:
                System.out.println("Exiting the program.");
                break;
              default:
    System.out.println("Invalid choice. Please enter a valid option.");
          }
        } while (choice != 3);
    }
}
```



```
Menu:
1. Display Student Details
2. Display Teacher Details
3. Exit
Enter your choice: 1

Student Details:
Name: Diksh
Age: 20
Student ID: 12345

Menu:
1. Display Student Details
2. Display Teacher Details
3. Exit
Enter your choice: 3
Exiting the program.
```

25. Create a class for employee having eno,ename and esal as data members. Provide functions for reading and displaying employee details. (Accept information of n employees in the main function, display the same and search for an emp (using eno)).

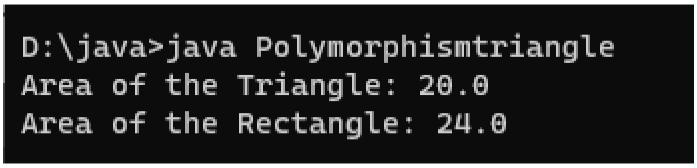
```
import java.util.Scanner;
class Employee {
   private int eno;
   private String ename;
   private double esal;
  public void readDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Employee Number: ");
        eno = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character
        System.out.print("Enter Employee Name: ");
        ename = scanner.nextLine();
        System.out.print("Enter Employee Salary: ");
        esal = scanner.nextDouble();
    }
  public void displayDetails() {
        System.out.println("Employee Number: " + eno);
        System.out.println("Employee Name: " + ename);
        System.out.println("Employee Salary: " + esal);
    }
        public boolean isEmployee(int searchEno) {
        return eno == searchEno;
    }
}
public class Employees {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
       System.out.print("Enter the number of employees: ");
       int n = scanner.nextInt();
       Employee[] employees = new Employee[n];
       for (int i = 0; i < n; i++) {
    System.out.println("\nEnter details for Employee " + (i + 1) + ":");
            employees[i] = new Employee();
            employees[i].readDetails();
        }
        System.out.println("\nDetails of Employees:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nEmployee " + (i + 1) + ":");
            employees[i].displayDetails();
        }
      System.out.print("\nEnter Employee Number to search: ");
        int searchEno = scanner.nextInt();
        boolean found = false;
        for (int i = 0; i < n; i++) {
            if (employees[i].isEmployee(searchEno)) {
                System.out.println("\nEmployee found. Details:");
                employees[i].displayDetails();
                found = true;
                break;
            }
        }
        if (!found) {
            System.out.println("\nEmployee not found with Employee Number: " + searchEno);
        }
  }
}
```



```
Enter details for Employee 1:
Enter Employee Number: 10001
Enter Employee Name: atlas
Enter Employee Salary: 200000

Enter details for Employee 2:
Enter Employee Number: 10002
Enter Employee Name: lily
Enter Employee Salary: 300000

Details of Employees:

Employee 1:
Employee Number: 10001
Employee Name: atlas
Employee Salary: 200000.0

Employee 2:
Employee Number: 10002
Employee Name: lily
Employee Salary: 300000.0

Enter Employee Number to search: 1

Employee not found with Employee Number: 1
```

26. Program to implement run time polymorphism in Java using interface, wrt calculating area of a triangle.

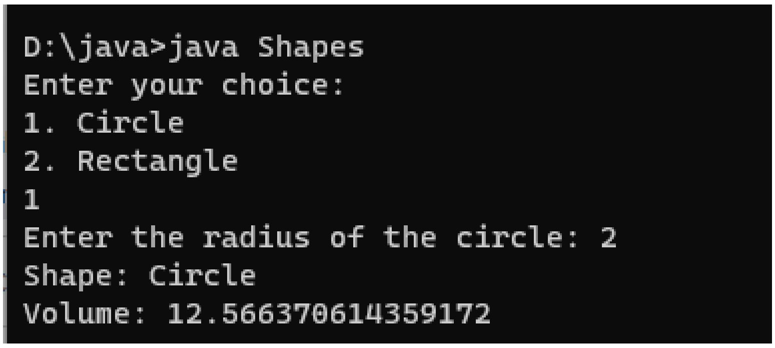
```
interface Shape {
  double calculateArea();
}
class Triangle implements Shape {
  private double base;
  private double height;
  public Triangle(double base, double height) {
    this.base = base;
    this.height = height;
  }
    public double calculateArea() {
    return 0.5 * base * height;
  }
}
class Rectangle implements Shape {
  private double length;
  private double width;
   public Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
  }
  public double calculateArea() {
    return length * width;
    }
}
public class Polymorphismtriangle {
    public static void main(String[] args) {
        Shape triangle = new Triangle(5, 8);
        Shape rectangle = new Rectangle(4, 6);
        System.out.println("Area of the Triangle: " + triangle.calculateArea());
        System.out.println("Area of the Rectangle: " + rectangle.calculateArea());
    }
}
```



```
D:\java>java Polymorphismtriangle
Area of the Triangle: 20.0
Area of the Rectangle: 24.0
```

27. Create an interface Shape having two prototypes disp() and calc(), to diplay the shape and calculate volume respectively. Create two classes circle and rectangle which implements the above interface. In the main function create a reference of Shape depending on the user-choice.

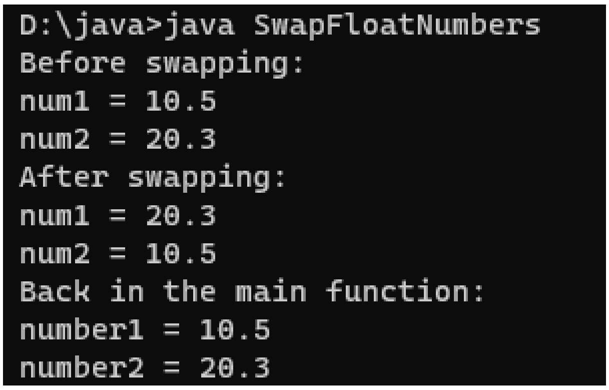
```
import java.util.Scanner;
interface Shape {
    void disp(); // Prototype to display the shape
    void calc(); // Prototype to calculate volume
}
class Circle implements Shape {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
        public void disp() {
        System.out.println("Shape: Circle");
    }
    public void calc() {
        double volume = Math.PI * radius * radius;
        System.out.println("Volume: " + volume);
    }
}
class Rectangle implements Shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
        public void disp() {
        System.out.println("Shape: Rectangle");
    }
    public void calc() {
        double volume = length * width;
        System.out.println("Volume: " + volume);
    }
}
public class Shapes {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter your choice:");
        System.out.println("1. Circle");
        System.out.println("2. Rectangle");
        int choice = scanner.nextInt();
        Shape shape = null;
       switch (choice) {
            case 1:
                System.out.print("Enter the radius of the circle: ");
                double radius = scanner.nextDouble();
                shape = new Circle(radius);
                break;
            case 2:
                System.out.print("Enter the length of the rectangle: ");
                double length = scanner.nextDouble();
                System.out.print("Enter the width of the rectangle: ");
                double width = scanner.nextDouble();
                shape = new Rectangle(length, width);
                break;
            default:
                System.out.println("Invalid choice.");
        }
       if (shape != null) {
            shape.disp();
            shape.calc();
        }
    }
}
```



```
D:\java>java Shapes
Enter your choice:
1. Circle
2. Rectangle
1
Enter the radius of the circle: 2
Shape: Circle
Volume: 12.566370614359172
```

27. Write a program to implement a function using call by value to swap two float numbers.

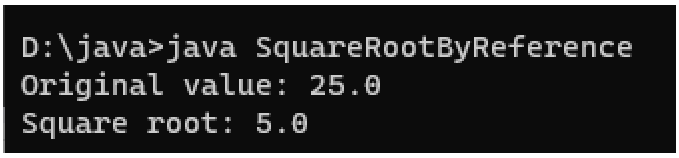
```
public class SwapFloatNumbers {
    public static void swap(float num1, float num2) {
    System.out.println("Before swapping:");
    System.out.println("num1 = " + num1);
    System.out.println("num2 = " + num2);
    float temp = num1;
    num1 = num2;
    num2 = temp;
    System.out.println("After swapping:");
    System.out.println("num1 = " + num1);
    System.out.println("num2 = " + num2);
  }
  public static void main(String[] args) {
     float number1 = 10.5f;
    float number2 = 20.3f;
   swap(number1, number2);
    System.out.println("Back in the main function:");
    System.out.println("number1 = " + number1);
    System.out.println("number2 = " + number2);
  }
}
```



```
D:\java>java SwapFloatNumbers
Before swapping:
num1 = 10.5
num2 = 20.3
After swapping:
num1 = 20.3
num2 = 10.5
Back in the main function:
number1 = 10.5
number2 = 20.3
```

28. Write a program to implement a function using call by reference to find the square root of a given number.

```
class NumberContainer {
  double value;
  public NumberContainer(double value) {
    this.value = value;
  }
}
public class SquareRootByReference {
    public static void findSquareRoot(NumberContainer numContainer) {
    numContainer.value = Math.sqrt(numContainer.value);
  }
  public static void main(String[] args) {
  NumberContainer numContainer = new NumberContainer(25.0);
  System.out.println("Original value: " + numContainer.value);
  findSquareRoot(numContainer);
  System.out.println("Square root: " + numContainer.value);
  }
}
```

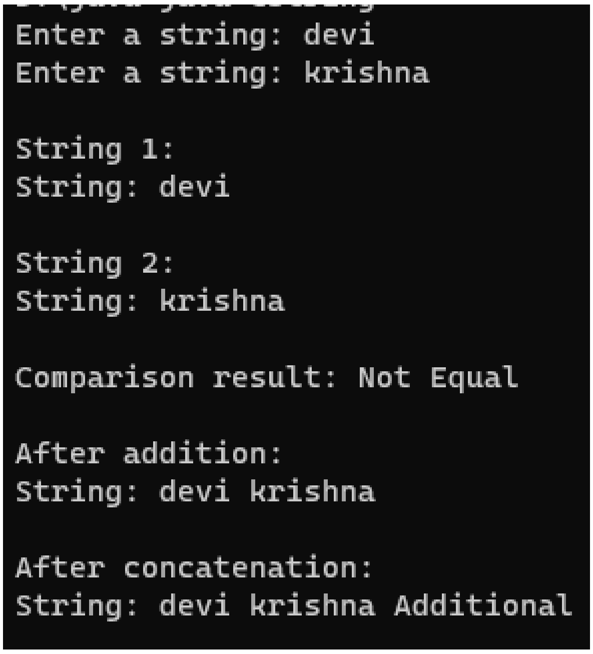


```
D:\java>java SquareRootByReference
Original value: 25.0
Square root: 5.0
```

29.Create a class for Cstring having a string data member and provide functions for read,display,compare(return Boolean value),add and concatenate.

```
 import java.util.Scanner;
public class CString {
  private String str;
   public void readString() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a string: ");
    str = scanner.nextLine();
  }
   public void displayString() {
    if (str != null) {
      System.out.println("String: " + str);
    } else {
      System.out.println("String is empty.");
    }
  }
    public boolean compareStrings(CString other) {
    return str.equals(other.str);
  }
   public void addString(CString other) {
    if (str != null && other.str != null) {
      str += other.str;
    } else {
      System.out.println("Both strings should be non-empty for addition.");
    }
  }
    public void concatenateString(String additionalStr) {
    if (additionalStr != null) {
      str += additionalStr;
    } else {
 System.out.println("The additional string should be non-empty for concatenation.");
    }
  }
  public static void main(String[] args) {
    CString string1 = new CString();
    CString string2 = new CString();
    string1.readString();
```

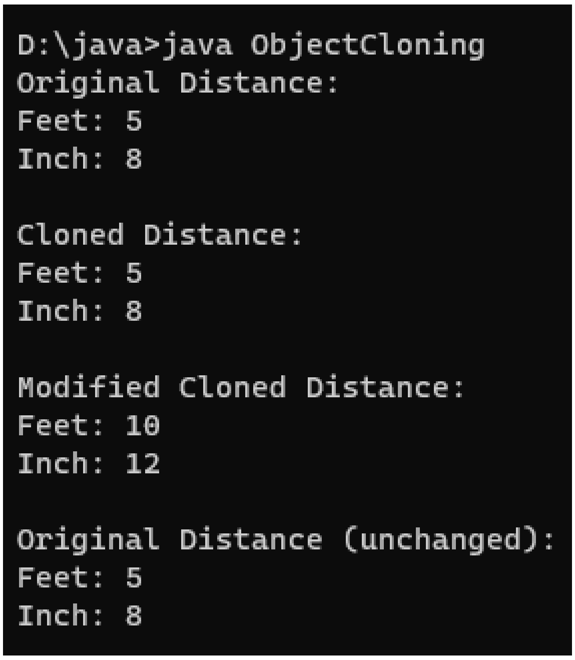
```
        string2.readString();
        System.out.println("\nString 1:");
        string1.displayString();
        System.out.println("\nString 2:");
        string2.displayString();
 System.out.println("\nComparison result: " + (string1.compareStrings(string2) ? "Equal" : "Not
Equal"));
        string1.addString(string2);
        System.out.println("\nAfter addition:");
        string1.displayString();
        String additionalStr = " Additional";
        string1.concatenateString(additionalStr);
        System.out.println("\nAfter concatenation:");
        string1.displayString();
    }
}
```



```
Enter a string: devi
Enter a string: krishna

String 1:
String: devi

String 2:
String: krishna

Comparison result: Not Equal

After addition:
String: devi krishna

After concatenation:
String: devi krishna Additional
```

30. Write a program to implement object cloning for the class Distance which has inch and feet as data members.

```
class Distance implements Cloneable {
    private int feet;
    private int inch;
    public Distance(int feet, int inch) {
        this.feet = feet;
        this.inch = inch;
    }
    public int getFeet() {
        return feet;
    }
    public int getInch() {
        return inch;
    }
    public Object clone() throws CloneNotSupportedException {
        return super.clone();
    }
}
public class ObjectCloning {
    public static void main(String[] args) {
        Distance originalDistance = new Distance(5, 8);
        System.out.println("Original Distance:");
        System.out.println("Feet: " + originalDistance.getFeet());
        System.out.println("Inch: " + originalDistance.getInch());
        try {
            Distance clonedDistance = (Distance) originalDistance.clone();
            System.out.println("\nCloned Distance:");
            System.out.println("Feet: " + clonedDistance.getFeet());
            System.out.println("Inch: " + clonedDistance.getInch());
            Distance modifiedClonedDistance = new Distance(10, 12);
            System.out.println("\nModified Cloned Distance:");
            System.out.println("Feet: " + modifiedClonedDistance.getFeet());
            System.out.println("Inch: " + modifiedClonedDistance.getInch());
            System.out.println("\nOriginal Distance (unchanged):");
            System.out.println("Feet: " + originalDistance.getFeet());
            System.out.println("Inch: " + originalDistance.getInch());
        } catch (CloneNotSupportedException e) {
            e.printStackTrace();
        }
    }
}
```

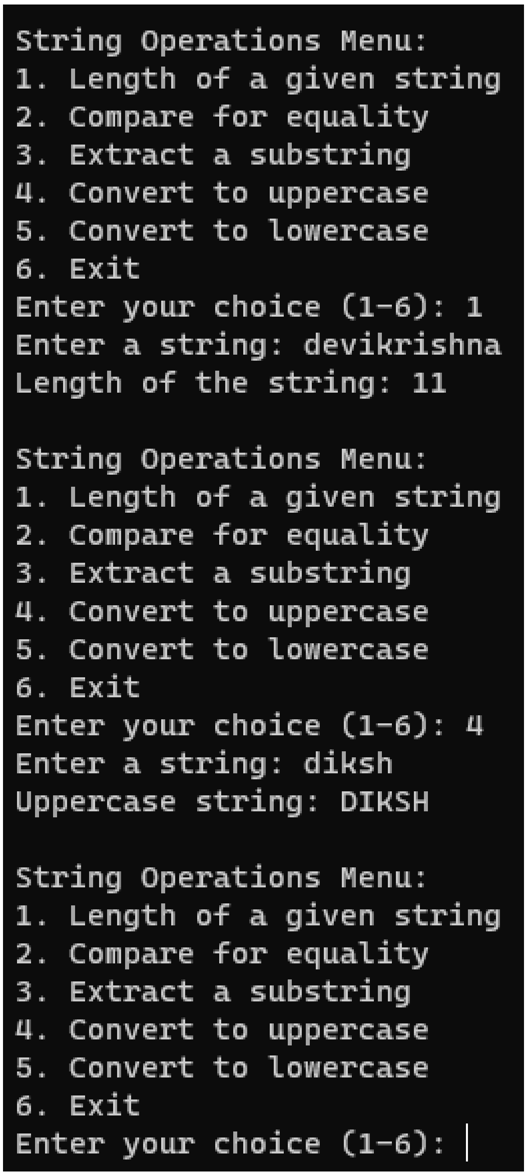


```
D:\java>java ObjectCloning
Original Distance:
Feet: 5
Inch: 8

Cloned Distance:
Feet: 5
Inch: 8

Modified Cloned Distance:
Feet: 10
Inch: 12

Original Distance (unchanged):
Feet: 5
Inch: 8
```

31. Write a program to create a menu driven program for performing the following operations. · Length of a given string · Compare for equality · Extract a substring from a string. · Convert to uppercase and lowercase.

```
import java.util.Scanner;
```

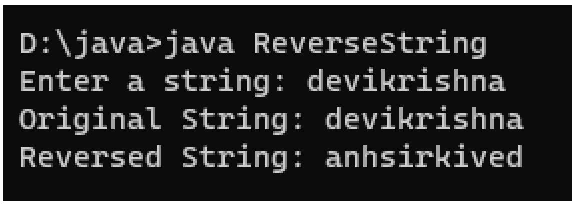
```
public class StringOperationsMenu {
public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("\nString Operations Menu:");
            System.out.println("1. Length of a given string");
            System.out.println("2. Compare for equality");
            System.out.println("3. Extract a substring");
            System.out.println("4. Convert to uppercase");
            System.out.println("5. Convert to lowercase");
            System.out.println("6. Exit");
            System.out.print("Enter your choice (1-6): ");
            int choice = scanner.nextInt();
            scanner.nextLine();
            switch (choice) {
                case 1:
                     System.out.print("Enter a string: ");
                    String inputString = scanner.nextLine();
                    int length = inputString.length();
                    System.out.println("Length of the string: " + length);
                    break;
                case 2:
                    System.out.print("Enter the first string: ");
                    String str1 = scanner.nextLine();
                    System.out.print("Enter the second string: ");
                    String str2 = scanner.nextLine();
                    boolean isEqual = str1.equals(str2);
           System.out.println("The strings are " + (isEqual ? "equal." : "not equal."));
                    break;
                case 3:
                    System.out.print("Enter the main string: ");
                    String mainString = scanner.nextLine();
                    System.out.print("Enter the starting index: ");
                    int startIndex = scanner.nextInt();
                    System.out.print("Enter the ending index: ");
                    int endIndex = scanner.nextInt();
                    String substring = mainString.substring(startIndex, endIndex);
                    System.out.println("Extracted substring: " + substring);
                    break;
                case 4:
                    System.out.print("Enter a string: ");
                    String upperCaseString = scanner.nextLine();
                    String upperCaseResult = upperCaseString.toUpperCase();
                    System.out.println("Uppercase string: " + upperCaseResult);
                    break;
                case 5:
                    System.out.print("Enter a string: ");
                    String lowerCaseString = scanner.nextLine();
                    String lowerCaseResult = lowerCaseString.toLowerCase();
                    System.out.println("Lowercase string: " + lowerCaseResult);
                    break;
                case 6:
                    System.out.println("Exiting the program. Goodbye!");
                    System.exit(0);
                default:
              System.out.println("Invalid choice. Please enter a number between 1 and 6.");
            }
        }
    }
}
```



```
String Operations Menu:
1. Length of a given string
2. Compare for equality
3. Extract a substring
4. Convert to uppercase
5. Convert to lowercase
6. Exit
Enter your choice (1-6): 1
Enter a string: devikrishna
Length of the string: 11

String Operations Menu:
1. Length of a given string
2. Compare for equality
3. Extract a substring
4. Convert to uppercase
5. Convert to lowercase
6. Exit
Enter your choice (1-6): 4
Enter a string: diksh
Uppercase string: DIKSH

String Operations Menu:
1. Length of a given string
2. Compare for equality
3. Extract a substring
4. Convert to uppercase
5. Convert to lowercase
6. Exit
Enter your choice (1-6): 
```

32. Write a program to reverse a string

```
import java.util.Scanner;
public class ReverseString {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String originalString = scanner.nextLine();
        String reversedString = reverseString(originalString);
        System.out.println("Original String: " + originalString);
        System.out.println("Reversed String: " + reversedString);
        scanner.close();
    }
     private static String reverseString(String str) {
        char[] charArray = str.toCharArray();
        int start = 0;
        int end = charArray.length - 1;
        while (start < end) {
           char temp = charArray[start];
          charArray[start] = charArray[end];
          charArray[end] = temp;
           start++;
          end--;
        }
        return new String(charArray);
    }
}
```

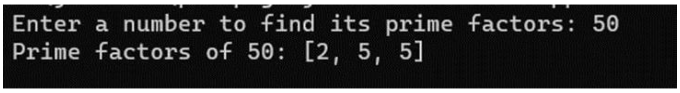


```
D:\java>java ReverseString
Enter a string: devikrishna
Original String: devikrishna
Reversed String: anhsirkived
```

33.Write a program to calculate the prime factors of a given number, using packages.

```
package mathutils;
import java.util.ArrayList;
import java.util.List;
public class PrimeFactorsCalculator {
 public static List<Integer> calculatePrimeFactors(int number) {
    List<Integer> primeFactors = new ArrayList<Integer>();
    for (int i = 2; i <= number; i++) {
      while (number % i == 0) {
        primeFactors.add(i);
        number /= i;
      }
    }
return primeFactors;
  }
}
```

```
package mathutils;
public class CombinationCalculator {
  private static long factorial(int n) {
      if (n == 0 || n == 1) {
         return 1;
      } else {
         return n * factorial(n - 1);
      }
   }
public static long calculateCombination(int n, int r) {
      if (n < r) {
         return 0; // Invalid input, as n should be greater than or equal to r
      }
      long numerator = factorial(n);
      long denominator = factorial(r) * factorial(n - r);
      return numerator / denominator;
   }
}
```

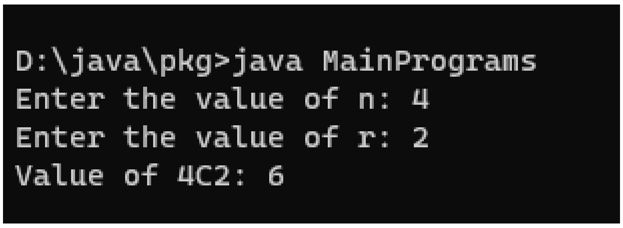


```
Enter a number to find its prime factors: 50
Prime factors of 50: [2, 5, 5]
```

34. Write a program to calculate the value of nCr for given value of n & r, using packages.

```
package combinatorics;

public class CombinationCalculator {
  public static long calculateCombination(int n, int r) {
      if (r < 0 || r > n) {
         return 0; // Invalid values for nCr
      }
      long numerator = calculateFactorial(n);
      long denominator = calculateFactorial(r) * calculateFactorial(n - r);
      return numerator / denominator;
   }
   private static long calculateFactorial(int number) {
      if (number == 0 || number == 1) {
         return 1;
      } else {
         return number * calculateFactorial(number - 1);
      }
   }
}

import combinatorics.CombinationCalculator;
import java.util.Scanner;
public class MainPrograms {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
     System.out.print("Enter the value of n: ");
      int n = scanner.nextInt();
      System.out.print("Enter the value of r: ");
```

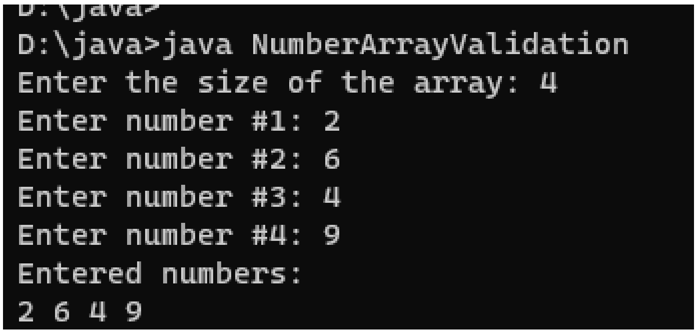
```
        int r = scanner.nextInt();
        long result = CombinationCalculator.calculateCombination(n, r);
        System.out.println("Value of " + n + "C" + r + ": " + result);
        scanner.close();
    }
}
```



```
D:\java\pkg>java MainPrograms
Enter the value of n: 4
Enter the value of r: 2
Value of 4C2: 6
```

35. Read numbers into an array. Perform validations using multiple catch statements / predefined Exceptions.

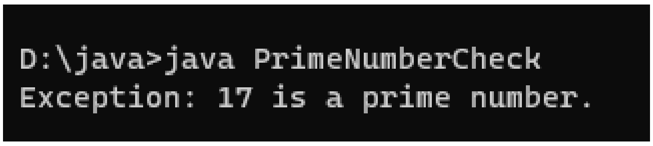
```
import java.util.*;

public class NumberArrayValidation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter the size of the array: ");
            int size = scanner.nextInt();

            if (size <= 0) {
                throw new IllegalArgumentException("Array size must be greater than 0");
            }
            int[] numbers = new int[size];

            for (int i = 0; i < size; i++) {
                System.out.print("Enter number #" + (i + 1) + ": ");
                numbers[i] = scanner.nextInt();
            }

            System.out.println("Entered numbers:");
            for (int num : numbers) {
                System.out.print(num + " ");
            }

        } catch (InputMismatchException e) {
            System.out.println("Error: Please enter a valid integer.");
            scanner.nextLine(); // consume the invalid input
        } catch (IllegalArgumentException e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
            scanner.close();
        }
    }
}
```



```
D:\java>
D:\java>java NumberArrayValidation
Enter the size of the array: 4
Enter number #1: 2
Enter number #2: 6
Enter number #3: 4
Enter number #4: 9
Entered numbers:
2 6 4 9
```

36. Write a program to implement a user defined Exception, which will throw an Exception when a given number is prime.

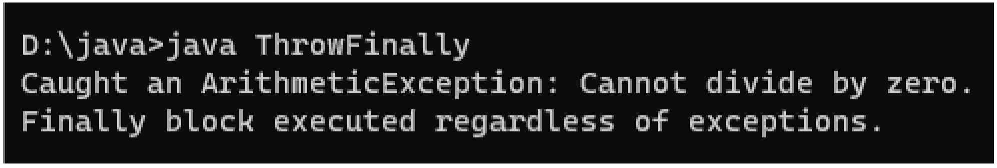
```
class PrimeNumberException extends Exception {
    public PrimeNumberException(String message) {
        super(message);
    }
}

public class PrimeNumberCheck {
    public static void main(String[] args) {
        try {
            int numberToCheck = 17; // Replace this with the number you want to check

            if (isPrime(numberToCheck)) {
                throw new PrimeNumberException(numberToCheck + " is a prime number.");
            } else {
                System.out.println(numberToCheck + " is not a prime number.");
            }
        } catch (PrimeNumberException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }

    private static boolean isPrime(int number) {
        if (number <= 1) {
            return false;
        }

        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```



```
D:\java>java PrimeNumberCheck
Exception: 17 is a prime number.
```

37. Write a program to implement throw and finally.

```
public class ThrowFinally {
    public static void main(String[] args) {
        try {
            // Simulating an error condition
            int result = divide(10, 0);
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Caught an ArithmeticException: " + e.getMessage());
        } finally {
            System.out.println("Finally block executed regardless of exceptions.");
        }
    }

    private static int divide(int numerator, int denominator) {
        if (denominator == 0) {
            throw new ArithmeticException("Cannot divide by zero.");
        }
        return numerator / denominator;
    }
}
```

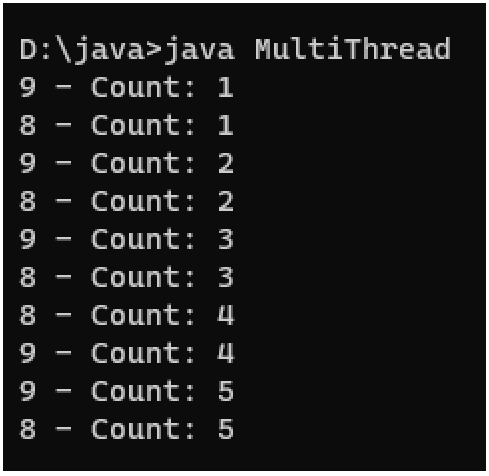


```
D:\java>java ThrowFinally
Caught an ArithmeticException: Cannot divide by zero.
Finally block executed regardless of exceptions.
```

38. Write a program to create multiple threads by extending the Thread class.

```
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(Thread.currentThread().getId() + " - Count: " + i);
            try {
                Thread.sleep(1000); // Sleep for 1 second
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}

public class MultiThread {
    public static void main(String[] args) {
        MyThread thread1 = new MyThread();
        MyThread thread2 = new MyThread();

        thread1.start(); // Start the first thread
        thread2.start(); // Start the second thread
```

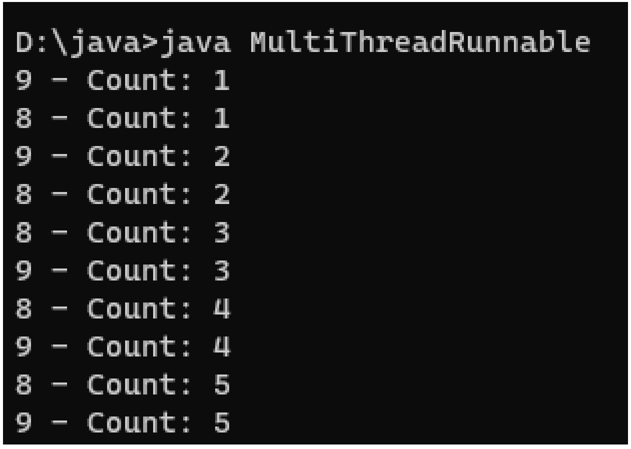
```
   }
}
```



```
D:\java>java MultiThread
9 - Count: 1
8 - Count: 1
9 - Count: 2
8 - Count: 2
9 - Count: 3
8 - Count: 3
8 - Count: 4
9 - Count: 4
9 - Count: 5
8 - Count: 5
```

39. Write a program to implement threads by implementing the Runnable interface

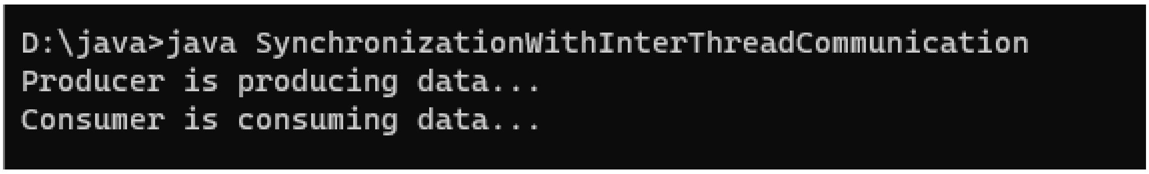
```
class MyRunnable implements Runnable {
   public void run() {
      for (int i = 1; i <= 5; i++) {
         System.out.println(Thread.currentThread().getId() + " - Count: " + i);
         try {
            Thread.sleep(1000); // Sleep for 1 second
         } catch (InterruptedException e) {
            System.out.println(e.getMessage());
         }
      }
   }
}
public class MultiThreadRunnable {
   public static void main(String[] args) {
      Thread thread1 = new Thread(new MyRunnable());
      Thread thread2 = new Thread(new MyRunnable());
      thread1.start(); // Start the first thread
      thread2.start(); // Start the second thread
   }
}
```



```
D:\java>java MultiThreadRunnable
9 - Count: 1
8 - Count: 1
9 - Count: 2
8 - Count: 2
8 - Count: 3
9 - Count: 3
8 - Count: 4
9 - Count: 4
8 - Count: 5
9 - Count: 5
```

40.Write a program to implement Synchronization using inter-thread communication.

```
class SharedResource {
    private boolean isDataReady = false;
    public synchronized void produceData() {
        System.out.println("Producer is producing data...");
        isDataReady = true;
        notify();
    }
    public synchronized void consumeData() {
        try {
            while (!isDataReady) {
                System.out.println("Consumer is waiting for data...");
                wait();
            }
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
        System.out.println("Consumer is consuming data...");
        isDataReady = false;
    }
}
class Producer extends Thread {
    private SharedResource sharedResource;
    public Producer(SharedResource sharedResource) {
        this.sharedResource = sharedResource;
    }
    public void run() {
        sharedResource.produceData();
    }
}
class Consumer extends Thread {
    private SharedResource sharedResource;
    public Consumer(SharedResource sharedResource) {
        this.sharedResource = sharedResource;
    }
    public void run() {
        sharedResource.consumeData();
    }
}
public class SynchronizationWithInterThreadCommunication {
    public static void main(String[] args) {
        SharedResource sharedResource = new SharedResource();
        Producer producerThread = new Producer(sharedResource);
        Consumer consumerThread = new Consumer(sharedResource);
        producerThread.start(); // Start the producer thread
        consumerThread.start(); // Start the consumer thread
    }
}
```



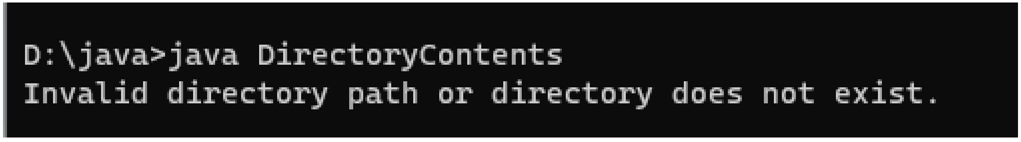
```
D:\java>java SynchronizationWithInterThreadCommunication
Producer is producing data...
Consumer is consuming data...
```

41. Implement the Producer- Consumer Problem, using Threads.

```
import java.util.LinkedList;


class SharedBuffer {
    private LinkedList<Integer> buffer = new LinkedList<>();
    private int capacity;

    public SharedBuffer(int capacity) {
        this.capacity = capacity;
    }
    public synchronized void produce(int item) {
        while (buffer.size() == capacity) {
            try {
                // Buffer is full, wait for space
                wait();
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        buffer.add(item);
        System.out.println("Produced: " + item);
        notify(); // Notify consumers that an item is produced
    }
    public synchronized void consume() {
        while (buffer.isEmpty()) {
            try {
                // Buffer is empty, wait for items
                wait();
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        int item = buffer.remove();
        System.out.println("Consumed: " + item);
        notify(); // Notify producers that space is available
    }
}
class Producer extends Thread {
    private SharedBuffer sharedBuffer;
    public Producer(SharedBuffer sharedBuffer) {
        this.sharedBuffer = sharedBuffer;
    }
    public void run() {
        for (int i = 1; i <= 5; i++) {
            sharedBuffer.produce(i);
            try {
                sleep(1000); // Sleep for 1 second
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
class Consumer extends Thread {
    private SharedBuffer sharedBuffer;
    public Consumer(SharedBuffer sharedBuffer) {
        this.sharedBuffer = sharedBuffer;
    }
    public void run() {
        for (int i = 1; i <= 5; i++) {
            sharedBuffer.consume();
            try {
                sleep(1000); // Sleep for 1 second
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
public class ProducerConsumer {
    public static void main(String[] args) {
        SharedBuffer sharedBuffer = new SharedBuffer(2); // Buffer size is 2
        Producer producerThread = new Producer(sharedBuffer);
        Consumer consumerThread = new Consumer(sharedBuffer);
        producerThread.start(); // Start the producer thread
        consumerThread.start(); // Start the consumer thread
    }
}
```

42. Write a program to display the contents of a directory by displaying the subdirectory’s name first, then the file names.

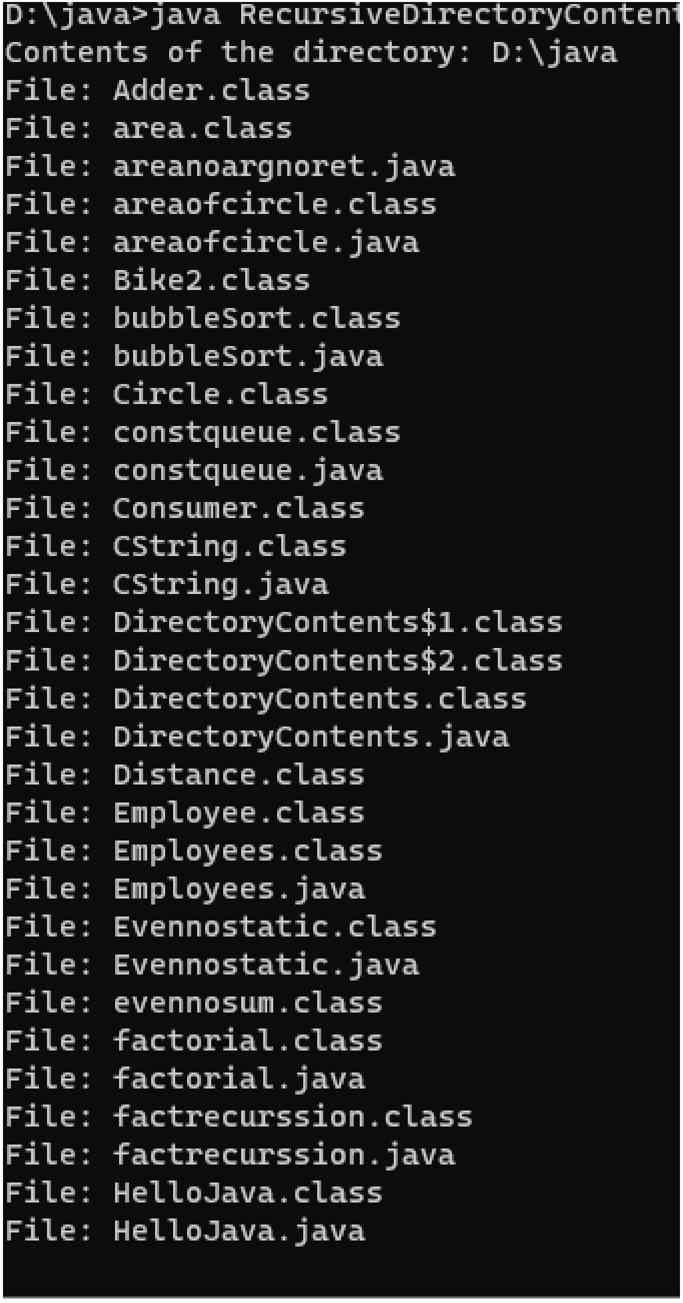
```
import java.io.File;
import java.io.FileFilter;

public class DirectoryContents {
    public static void main(String[] args) {
            String directoryPath = "D:\\example_directory";
        File directory = new File(directoryPath);
        if (directory.exists() && directory.isDirectory()) {
            System.out.println("Contents of the directory: " + directoryPath);

            File[] subdirectories = directory.listFiles(new FileFilter() {

                public boolean accept(File file) {
                    return file.isDirectory();
                }
            });

            if (subdirectories != null) {
                System.out.println("Subdirectories:");
                for (File subdirectory : subdirectories) {
                    System.out.println(subdirectory.getName());
                }
            }
            File[] files = directory.listFiles(new FileFilter() {
             public boolean accept(File file) {
                    return file.isFile();
                }
            });

            if (files != null) {
                System.out.println("Files:");
                for (File file : files) {
                    System.out.println(file.getName());
                }
            }
        } else {
            System.out.println("Invalid directory path or directory does not exist.");
        }
    }
}
```



```
D:\java>java DirectoryContents
Invalid directory path or directory does not exist.
```

43.Write a program to display the contents of a directory including its subdirectory content using recursive function.

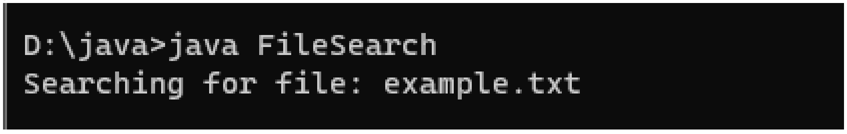
```
import java.io.File;
public class RecursiveDirectoryContents {
    public static void main(String[] args) {
        // Replace the path with the directory path you want to display
        String directoryPath = "D:\\java";
        File directory = new File(directoryPath);
        if (directory.exists() && directory.isDirectory()) {
            System.out.println("Contents of the directory: " + directoryPath);
            displayDirectoryContents(directory);
        } else {
            System.out.println("Invalid directory path or directory does not exist.");
        }
    }
    private static void displayDirectoryContents(File directory) {
        File[] files = directory.listFiles();
        if (files != null) {
            for (File file : files) {
                if (file.isDirectory()) {
                    // If it's a directory, display its contents recursively
                    System.out.println("Subdirectory: " + file.getName());
                    displayDirectoryContents(file);
                } else {
                    // If it's a file, display its name
                    System.out.println("File: " + file.getName());
                }
            }
        }
    }
}
```



```
D:\java>java RecursiveDirectoryContent
Contents of the directory: D:\java
File: Adder.class
File: area.class
File: areanoargnoret.java
File: areaofcircle.class
File: areaofcircle.java
File: Bike2.class
File: bubbleSort.class
File: bubbleSort.java
File: Circle.class
File: constqueue.class
File: constqueue.java
File: Consumer.class
File: CString.class
File: CString.java
File: DirectoryContents$1.class
File: DirectoryContents$2.class
File: DirectoryContents.class
File: DirectoryContents.java
File: Distance.class
File: Employee.class
File: Employees.class
File: Employees.java
File: Evennostatic.class
File: Evennostatic.java
File: evennosum.class
File: factorial.class
File: factorial.java
File: factrecurssion.class
File: factrecurssion.java
File: HelloJava.class
File: HelloJava.java
```

44. Write a program to search for a given file name in a directory (including its subdirectory) content.

```
import java.io.File;
public class FileSearch {
   public static void main(String[] args) {
      String directoryPath = "D:\\example_directory";
      String targetFileName = "example.txt";
      File directory = new File(directoryPath);
      if (directory.exists() && directory.isDirectory()) {
         System.out.println("Searching for file: " + targetFileName);
         searchForFile(directory, targetFileName);
      } else {
         System.out.println("Invalid directory path or directory does not exist.");
      }
   }
   private static void searchForFile(File directory, String targetFileName) {
      File[] files = directory.listFiles();
      if (files != null) {
         for (File file : files) {
            if (file.isDirectory()) {
               searchForFile(file, targetFileName);
            } else if (file.getName().equals(targetFileName)) {
               System.out.println("File found at: " + file.getAbsolutePath());
            }
         }
      }
   }
}
```



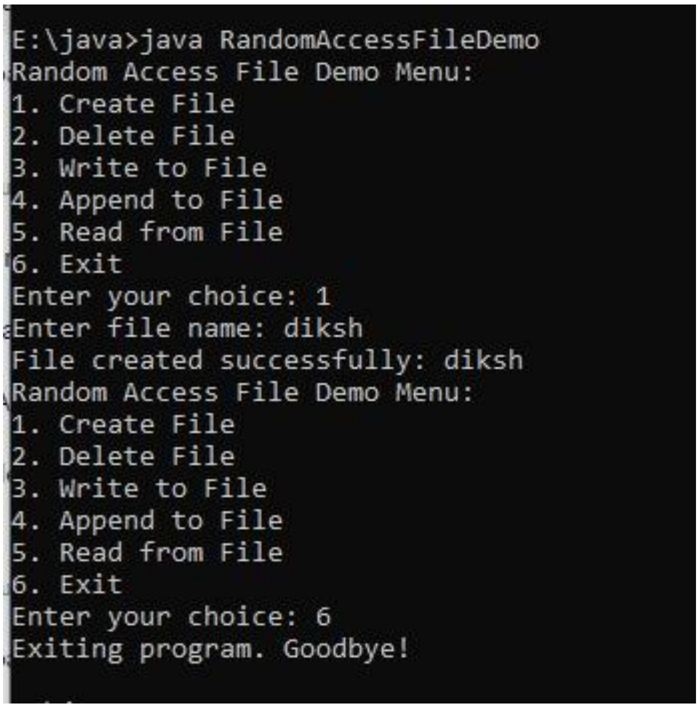
```
D:\java>java FileSearch
Searching for file: example.txt
```

45. Write a menu driven program to demonstrate Random Access File handling, with options for creating, deleting, writing, appending and reading the file.

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

public class RandomAccessFileDemo {


   public static void main(String[] args) {

      Scanner scanner = new Scanner(System.in);



      while (true) {

         System.out.println("Random Access File Demo Menu:");

         System.out.println("1. Create File");
```

```
        System.out.println("2. Delete File");
        System.out.println("3. Write to File");
        System.out.println("4. Append to File");
        System.out.println("5. Read from File");
        System.out.println("6. Exit");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        scanner.nextLine(); // consume the newline character
        switch (choice) {
            case 1:
                createFile();
                break;
            case 2:
                deleteFile();
                break;
            case 3:
                writeToFile();
                break;
            case 4:
                appendToFile();
                break;
            case 5:
                readFromFile();
                break;
            case 6:
                System.out.println("Exiting program. Goodbye!");
                System.exit(0);
            default:
                System.out.println("Invalid choice. Please enter a valid option.");
        }
    }
}
```

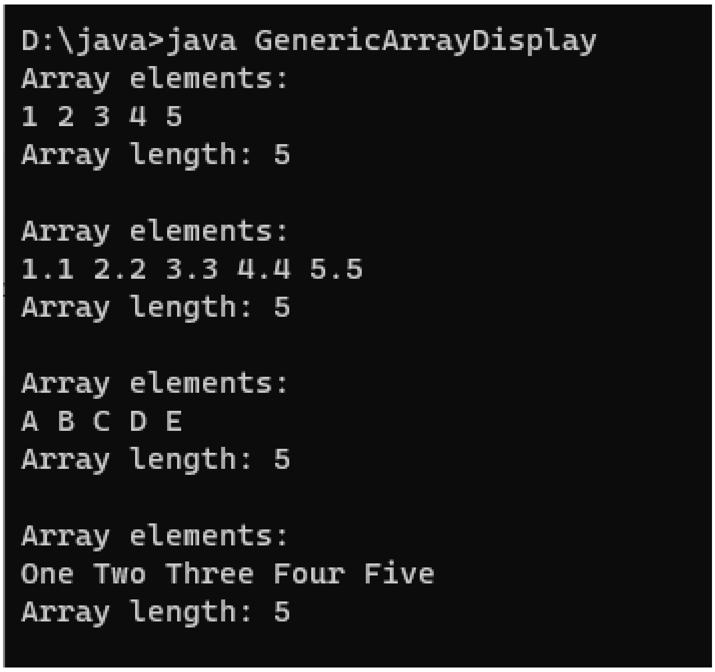
```
private static void createFile() {
    try {
        System.out.print("Enter file name: ");
        Scanner scanner = new Scanner(System.in);
        String fileName = scanner.nextLine();
        RandomAccessFile file = new RandomAccessFile(fileName, "rw");
        file.close();
        System.out.println("File created successfully: " + fileName);
    } catch (Exception e) {
        System.out.println("Error creating file: " + e.getMessage());
    }
}

private static void deleteFile() {
    try {
        System.out.print("Enter file name to delete: ");
        Scanner scanner = new Scanner(System.in);
        String fileName = scanner.nextLine();
        if (new java.io.File(fileName).delete()) {
            System.out.println("File deleted successfully: " + fileName);
        } else {
            System.out.println("File not found or could not be deleted.");
        }
    } catch (Exception e) {
        System.out.println("Error deleting file: " + e.getMessage());
    }
}

private static void writeToFile() {
    try {
        System.out.print("Enter file name to write: ");
        Scanner scanner = new Scanner(System.in);
        String fileName = scanner.nextLine();
        RandomAccessFile file = new RandomAccessFile(fileName, "rw");

        System.out.print("Enter data to write: ");
        String data = scanner.nextLine();
        file.writeUTF(data);

        file.close();
        System.out.println("Data written to file successfully: " + fileName);
      } catch (Exception e) {
        System.out.println("Error writing to file: " + e.getMessage());
      }
    }

    private static void appendToFile() {
      try {
        System.out.print("Enter file name to append: ");
        Scanner scanner = new Scanner(System.in);
        String fileName = scanner.nextLine();
        RandomAccessFile file = new RandomAccessFile(fileName, "rw");

        System.out.print("Enter data to append: ");
        String data = scanner.nextLine();

        file.seek(file.length());
        file.writeUTF(data);
        file.close();
  System.out.println("Data appended to file successfully: " + fileName);
      } catch (Exception e) {
        System.out.println("Error appending to file: " + e.getMessage());
      }
    }
    private static void readFromFile() {
      try {
        System.out.print("Enter file name to read: ");
        Scanner scanner = new Scanner(System.in);
        String fileName = scanner.nextLine();
        RandomAccessFile file = new RandomAccessFile(fileName, "r");

        while (file.getFilePointer() < file.length()) {
            String data = file.readUTF();
            System.out.println("Data read from file: " + data);
        }
        file.close();
      } catch (Exception e) {
        System.out.println("Error reading from file: " + e.getMessage());
      }
    }
  }
```



```
E:\java>java RandomAccessFileDemo
Random Access File Demo Menu:
1. Create File
2. Delete File
3. Write to File
4. Append to File
5. Read from File
6. Exit
Enter your choice: 1
Enter file name: diksh
File created successfully: diksh
Random Access File Demo Menu:
1. Create File
2. Delete File
3. Write to File
4. Append to File
5. Read from File
6. Exit
Enter your choice: 6
Exiting program. Goodbye!
```

46. Write a program to implement a Generic method, which can display the elements of various arrays of different data types, and find the length of each array.

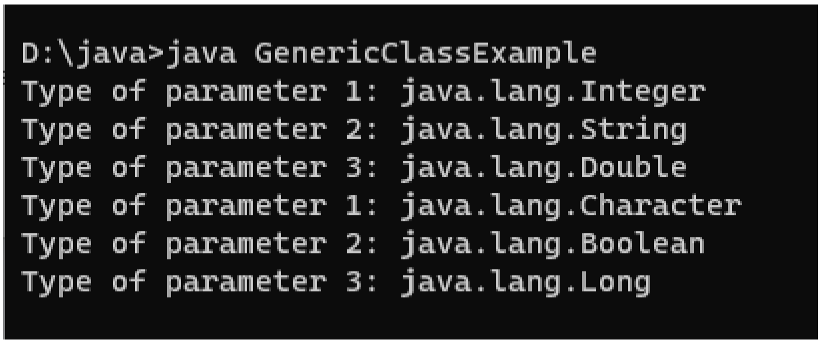
```
public class GenericArrayDisplay {
    public static <T> void displayArray(T[] array) {
        System.out.println("Array elements:");
        for (T element : array) {
            System.out.print(element + " ");
        }
        System.out.println("\nArray length: " + array.length);
        System.out.println();
    }
    public static void main(String[] args) {
            Integer[] intArray = {1, 2, 3, 4, 5};
        Double[] doubleArray = {1.1, 2.2, 3.3, 4.4, 5.5};
        Character[] charArray = {'A', 'B', 'C', 'D', 'E'};
        String[] stringArray = {"One", "Two", "Three", "Four", "Five"};
        displayArray(intArray);
        displayArray(doubleArray);
        displayArray(charArray);
        displayArray(stringArray);
    }
}
```



```
D:\java>java GenericArrayDisplay
Array elements:
1 2 3 4 5
Array length: 5

Array elements:
1.1 2.2 3.3 4.4 5.5
Array length: 5

Array elements:
A B C D E
Array length: 5

Array elements:
One Two Three Four Five
Array length: 5
```

47. Write a program to implement a Generic class, and display the types of various parameters passed.

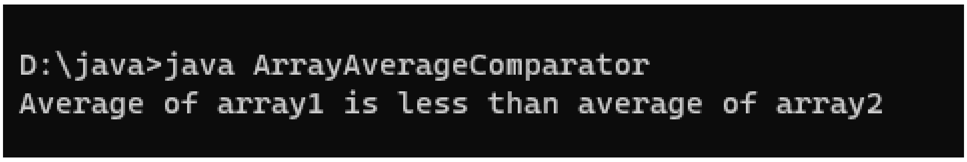
```
public class GenericClassExample<T, U, V> {
   private T type1;
   private U type2;
   private V type3;
   public GenericClassExample(T type1, U type2, V type3) {
      this.type1 = type1;
      this.type2 = type2;
      this.type3 = type3;
   }
   public void displayTypes() {
      System.out.println("Type of parameter 1: " + type1.getClass().getName());
      System.out.println("Type of parameter 2: " + type2.getClass().getName());
      System.out.println("Type of parameter 3: " + type3.getClass().getName());
   }
public static void main(String[] args) {
      GenericClassExample<Integer, String, Double> example1 = new
GenericClassExample<Integer, String, Double>(10, "Hello", 3.14);
   example1.displayTypes();
   GenericClassExample<Character, Boolean, Long> example2 = new
GenericClassExample<Character, Boolean, Long>('A', true, 100L);
   example2.displayTypes();
}
}
```



```
D:\java>java GenericClassExample
Type of parameter 1: java.lang.Integer
Type of parameter 2: java.lang.String
Type of parameter 3: java.lang.Double
Type of parameter 1: java.lang.Character
Type of parameter 2: java.lang.Boolean
Type of parameter 3: java.lang.Long
```

48. Using bounded type parameters, compare the averages of various arrays.

```
public class ArrayAverageComparator {
    public static <T extends Number> double calculateAverage(T[] array) {
        double sum = 0.0;

        for (T element : array) {

            sum += element.doubleValue();

        }

        return sum / array.length;

    }

      public static <T extends Number, U extends Number> String compareAverages(T[] array1,
U[] array2) {

        double average1 = calculateAverage(array1);

        double average2 = calculateAverage(array2);


        if (average1 > average2) {

            return "Average of array1 is greater than average of array2";

        } else if (average1 < average2) {

            return "Average of array1 is less than average of array2";

        } else {

            return "Averages of array1 and array2 are equal";

        }

    }

    public static void main(String[] args) {

             Integer[] intArray = {1, 2, 3, 4, 5};

        Double[] doubleArray = {1.5, 2.5, 3.5, 4.5, 5.5};

        String result = compareAverages(intArray, doubleArray);

        System.out.println(result);

    }
```



```
D:\java>java ArrayAverageComparator
Average of array1 is less than average of array2
```

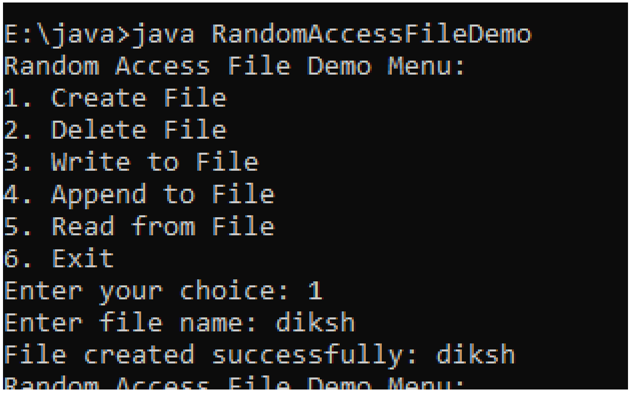
49. Write a menu driven program to demonstrate Random Access File handling, with options for creating, deleting, writing, appending and reading the file.

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

```java
public class RandomAccessFileDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("Random Access File Demo Menu:");
            System.out.println("1. Create File");
            System.out.println("2. Delete File");
            System.out.println("3. Write to File");
            System.out.println("4. Append to File");
            System.out.println("5. Read from File");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine();
            switch (choice) {
                case 1:
                    createFile();
                    break;
                case 2:
                    deleteFile();
                    break;
                case 3:
                    writeToFile();
                    break;
                case 4:
                    appendToFile();
                    break;
                case 5:
                    readFromFile();
                    break;
                case 6:
                    System.out.println("Exiting program. Goodbye!");
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please enter a valid option.");
            }
        }
    }
    private static void createFile() {
        try {
            System.out.print("Enter file name: ");
            Scanner scanner = new Scanner(System.in);
            String fileName = scanner.nextLine();
            RandomAccessFile file = new RandomAccessFile(fileName, "rw");
            file.close();
            System.out.println("File created successfully: " + fileName);
        } catch (Exception e) {
            System.out.println("Error creating file: " + e.getMessage());
        }
    }
```

```
private static void deleteFile() {
  try {
    System.out.print("Enter file name to delete: ");
    Scanner scanner = new Scanner(System.in);
    String fileName = scanner.nextLine();
    if (new java.io.File(fileName).delete()) {
      System.out.println("File deleted successfully: " + fileName);
    } else {
      System.out.println("File not found or could not be deleted.");
    }
  } catch (Exception e) {
    System.out.println("Error deleting file: " + e.getMessage());
  }
}
private static void writeToFile() {
  try {
    System.out.print("Enter file name to write: ");
    Scanner scanner = new Scanner(System.in);
    String fileName = scanner.nextLine();
    RandomAccessFile file = new RandomAccessFile(fileName, "rw");
    System.out.print("Enter data to write: ");
    String data = scanner.nextLine();
    file.writeUTF(data);
    file.close();
    System.out.println("Data written to file successfully: " + fileName);
  } catch (Exception e) {
    System.out.println("Error writing to file: " + e.getMessage());
  }
}
private static void appendToFile() {
  try {
    System.out.print("Enter file name to append: ");
    Scanner scanner = new Scanner(System.in);
    String fileName = scanner.nextLine();
    RandomAccessFile file = new RandomAccessFile(fileName, "rw");
    System.out.print("Enter data to append: ");
    String data = scanner.nextLine();
    file.seek(file.length());
    file.writeUTF(data);
    file.close();
    System.out.println("Data appended to file successfully: " + fileName);
  } catch (Exception e) {
    System.out.println("Error appending to file: " + e.getMessage());
  }
}
private static void readFromFile() {
  try {
    System.out.print("Enter file name to read: ");
    Scanner scanner = new Scanner(System.in);
    String fileName = scanner.nextLine();
```

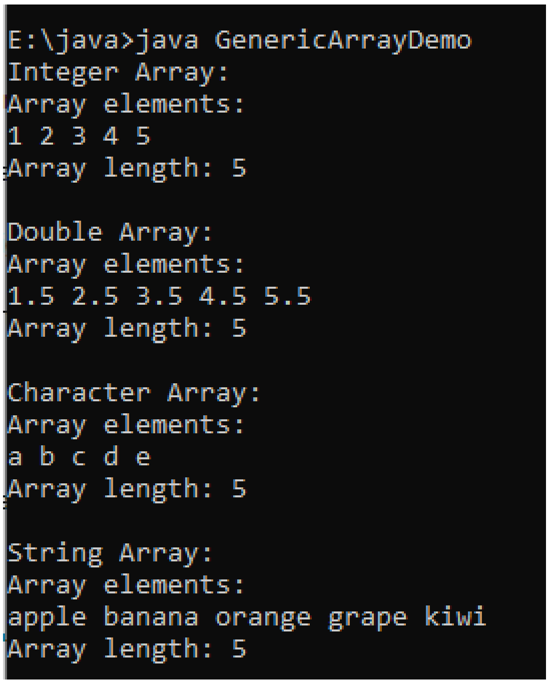
```
      RandomAccessFile file = new RandomAccessFile(fileName, "r");
      while (file.getFilePointer() < file.length()) {
        String data = file.readUTF();
        System.out.println("Data read from file: " + data);
      }
      file.close();
    } catch (Exception e) {
      System.out.println("Error reading from file: " + e.getMessage());
    }
  }
}
```



```
E:\java>java RandomAccessFileDemo
Random Access File Demo Menu:
1. Create File
2. Delete File
3. Write to File
4. Append to File
5. Read from File
6. Exit
Enter your choice: 1
Enter file name: diksh
File created successfully: diksh
Random Access File Demo Menu:
```

50. Write a program to implement a Generic method, which can display the elements of various arrays of different data types, and find the length of each array.

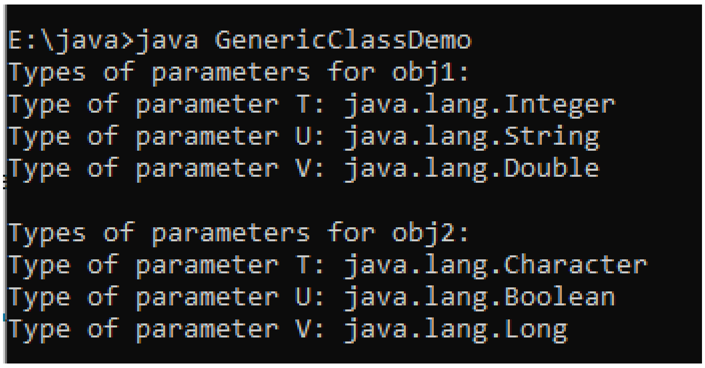
```
public class GenericArrayDemo {
    public static <T> void displayArrayInfo(T[] array) {
    System.out.println("Array elements:");
    for (T element : array) {
      System.out.print(element + " ");
    }
    System.out.println("\nArray length: " + array.length);
  }
  public static void main(String[] args) {
    Integer[] intArray = {1, 2, 3, 4, 5};
    Double[] doubleArray = {1.5, 2.5, 3.5, 4.5, 5.5};
    Character[] charArray = {'a', 'b', 'c', 'd', 'e'};
   String[] stringArray = {"apple", "banana", "orange", "grape", "kiwi"};

   System.out.println("Integer Array:");
    displayArrayInfo(intArray);

    System.out.println("\nDouble Array:");
    displayArrayInfo(doubleArray);
   System.out.println("\nCharacter Array:");
    displayArrayInfo(charArray);
    System.out.println("\nString Array:");
    displayArrayInfo(stringArray);
   }
}
```



```
E:\java>java GenericArrayDemo
Integer Array:
Array elements:
1 2 3 4 5
Array length: 5

Double Array:
Array elements:
1.5 2.5 3.5 4.5 5.5
Array length: 5

Character Array:
Array elements:
a b c d e
Array length: 5

String Array:
Array elements:
apple banana orange grape kiwi
Array length: 5
```

50. Write a program to implement a Generic class, and display the types of various parameters passed.

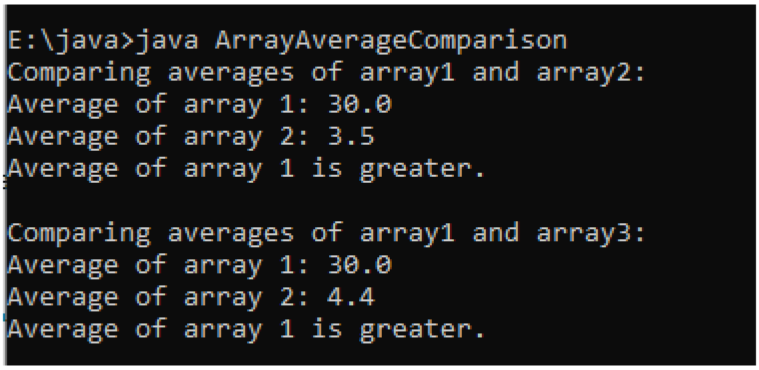
```
public class GenericClassDemo<T, U, V> {
   private T t;
   private U u;
   private V v;
   public GenericClassDemo(T t, U u, V v) {
      this.t = t;
      this.u = u;
      this.v = v;
   }
   public void displayTypes() {
      System.out.println("Type of parameter T: " + t.getClass().getName());
      System.out.println("Type of parameter U: " + u.getClass().getName());
      System.out.println("Type of parameter V: " + v.getClass().getName());
   }
   public static void main(String[] args) {
            GenericClassDemo<Integer, String, Double> obj1 = new GenericClassDemo<Integer, 
String, Double>(10, "Hello", 3.14);
      GenericClassDemo<Character, Boolean, Long> obj2 = new GenericClassDemo<Character, 
Boolean, Long>('A', true, 100L);
      System.out.println("Types of parameters for obj1:");
      obj1.displayTypes();
      System.out.println("\nTypes of parameters for obj2:");
      obj2.displayTypes();
   }
}
```



```
E:\java>java GenericClassDemo
Types of parameters for obj1:
Type of parameter T: java.lang.Integer
Type of parameter U: java.lang.String
Type of parameter V: java.lang.Double

Types of parameters for obj2:
Type of parameter T: java.lang.Character
Type of parameter U: java.lang.Boolean
Type of parameter V: java.lang.Long
```

52. Using bounded type parameters, compare the averages of various arrays.

```
public class ArrayAverageComparison {
    public static <T extends Number> void compareAverages(T[] array1, T[] array2) {
        double average1 = calculateAverage(array1);
        double average2 = calculateAverage(array2);
        System.out.println("Average of array 1: " + average1);
        System.out.println("Average of array 2: " + average2);
        if (average1 > average2) {
            System.out.println("Average of array 1 is greater.");
        } else if (average1 < average2) {
            System.out.println("Average of array 2 is greater.");
        } else {
            System.out.println("Averages of both arrays are equal.");
        }
    }
    private static <T extends Number> double calculateAverage(T[] array) {
        double sum = 0;
        for (T element : array) {
            sum += element.doubleValue();
        }
        return sum / array.length;
    }
    public static void main(String[] args) {
        Integer[] array1 = {10, 20, 30, 40, 50};
        Double[] array2 = {1.5, 2.5, 3.5, 4.5, 5.5};
        Float[] array3 = {2.2f, 3.3f, 4.4f, 5.5f, 6.6f};
        System.out.println("Comparing averages of array1 and array2:");
        compareAverages(array1, array2);
        System.out.println("\nComparing averages of array1 and array3:");
        compareAverages(array1, array3);
    }
}
```

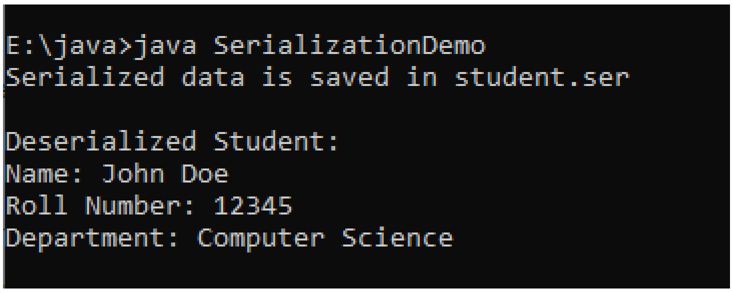


```
E:\java>java ArrayAverageComparison
Comparing averages of array1 and array2:
Average of array 1: 30.0
Average of array 2: 3.5
Average of array 1 is greater.

Comparing averages of array1 and array3:
Average of array 1: 30.0
Average of array 2: 4.4
Average of array 1 is greater.
```

53. Write a program to implement Serialization and De-Serialization, for an object of Student Class.

```
import java.io.*;

class Student implements Serializable {

  private static final long serialVersionUID = 1L;

  private String name;

  private int rollNumber;

  private String department;

  public Student(String name, int rollNumber, String department) {

    this.name = name;

    this.rollNumber = rollNumber;

    this.department = department;

  }

  public void display() {

    System.out.println("Name: " + name);

    System.out.println("Roll Number: " + rollNumber);

    System.out.println("Department: " + department);

  }

}

public class SerializationDemo {

  public static void main(String[] args) {

    try {

      Student student = new Student("John Doe", 12345, "Computer Science");

      FileOutputStream fileOut = new FileOutputStream("student.ser");

      ObjectOutputStream objectOut = new ObjectOutputStream(fileOut);

      objectOut.writeObject(student);
```

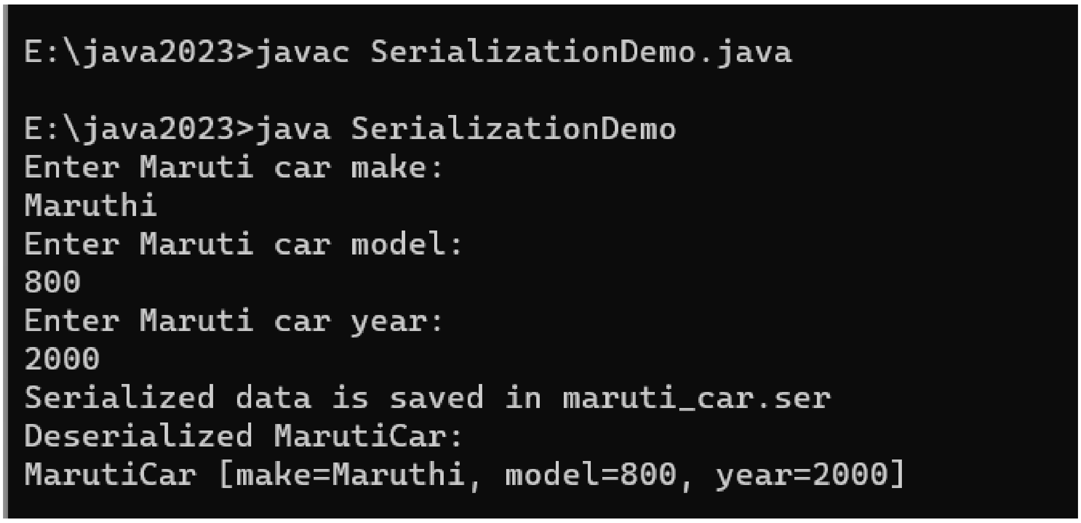
```
            objectOut.close();

            fileOut.close();

            System.out.println("Serialized data is saved in student.ser");

        } catch (IOException e) {

            e.printStackTrace();

        }

        try {

            FileInputStream fileIn = new FileInputStream("student.ser");

            ObjectInputStream objectIn = new ObjectInputStream(fileIn);

            Student student = (Student) objectIn.readObject();

            objectIn.close();

            fileIn.close();

            System.out.println("\nDeserialized Student:");

            student.display();

        } catch (IOException e) {

            e.printStackTrace();

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        }

    }

}
```



```
E:\java>java SerializationDemo
Serialized data is saved in student.ser

Deserialized Student:
Name: John Doe
Roll Number: 12345
Department: Computer Science
```

54. Write a program to implement IS-A Serialization and De-Serialization, for a Maruti Car inherited from Vehicle.

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

// Base class (parent)
class Vehicle implements Serializable {
    private static final long serialVersionUID = 1L;
    private String make;
    private String model;

    public Vehicle(String make, String model) {
        this.make = make;
        this.model = model;
    }

    public String getMake() {
        return make;
    }

    public String getModel() {
        return model;
    }

    @Override
    public String toString() {
        return "Vehicle [make=" + make + ", model=" + model + "]";
    }
}

// Derived class (child)
class MarutiCar extends Vehicle implements Serializable {
    private static final long serialVersionUID = 1L;
    private int year;

    public MarutiCar(String make, String model, int year) {
        super(make, model);
        this.year = year;
    }

    public int getYear() {
        return year;
    }
    public String toString() {
        return "MarutiCar [make=" + getMake() + ", model=" + getModel() + ", year=" + year + "]";
    }

}


public class SerializationDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Getting Maruti car details from user
        System.out.println("Enter Maruti car make:");
        String make = scanner.nextLine();

        System.out.println("Enter Maruti car model:");
        String model = scanner.nextLine();

        System.out.println("Enter Maruti car year:");
        int year = scanner.nextInt();

        MarutiCar car = new MarutiCar(make, model, year);

        // Serialization
        try {
            FileOutputStream fileOut = new FileOutputStream("maruti_car.ser");
            ObjectOutputStream out = new ObjectOutputStream(fileOut);
            out.writeObject(car);
            out.close();
            fileOut.close();
            System.out.println("Serialized data is saved in maruti_car.ser");
        } catch (IOException i) {
            i.printStackTrace();
        }


        // Deserialization
        MarutiCar deserializedCar = null;
        try {
            FileInputStream fileIn = new FileInputStream("maruti_car.ser");
            ObjectInputStream in = new ObjectInputStream(fileIn);
            deserializedCar = (MarutiCar) in.readObject();
            in.close();
            fileIn.close();
        } catch (IOException i) {
            i.printStackTrace();
            return;
        } catch (ClassNotFoundException c) {
            System.out.println("MarutiCar class not found");
            c.printStackTrace();
            return;
        }

        // Displaying deserialized car
        System.out.println("Deserialized MarutiCar:");
        System.out.println(deserializedCar);

        scanner.close();
    }

 }
```

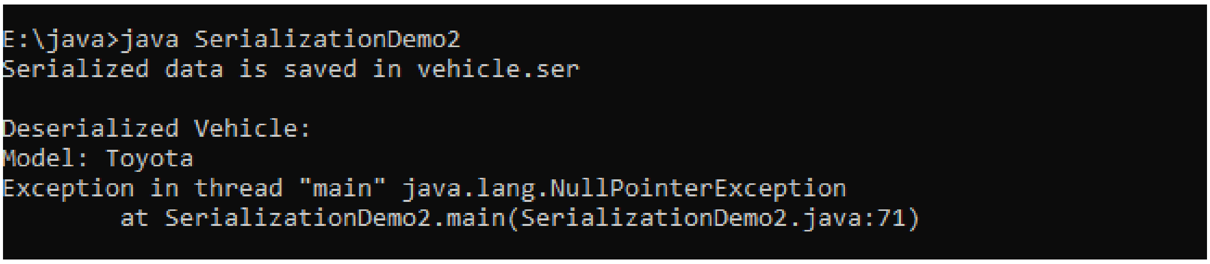


```
E:\java2023>javac SerializationDemo.java

E:\java2023>java SerializationDemo
Enter Maruti car make:
Maruthi
Enter Maruti car model:
800
Enter Maruti car year:
2000
Serialized data is saved in maruti_car.ser
Deserialized MarutiCar:
MarutiCar [make=Maruthi, model=800, year=2000]
```

55. Write a program to implement HAS-A Serialization and De-Serialization for the Engine of a Vehicle.

```
import java.io.*;

class Engine implements Serializable {

   private static final long serialVersionUID = 1L;

   private String type;

   public Engine(String type) {

      this.type = type;

   }

   public String getType() {

      return type;

   }

   public void setType(String type) {

      this.type = type;

   }
```

```
}
class Vehicle implements Serializable {
    private static final long serialVersionUID = 1L;
    private String model;
    private transient Engine engine; // transient keyword to exclude Engine from serialization
    public Vehicle(String model, String engineType) {
        this.model = model;
        this.engine = new Engine(engineType);
    }
    public String getModel() {
        return model;
    }
    public void setModel(String model) {
        this.model = model;
    }
    public Engine getEngine() {
        return engine;
    }
    public void setEngine(Engine engine) {
        this.engine = engine;
    }
}
public class SerializationDemo2 {
    public static void main(String[] args) {
        try {
            Vehicle car = new Vehicle("Toyota", "V8");
            FileOutputStream fileOut = new FileOutputStream("vehicle.ser");
            ObjectOutputStream objectOut = new ObjectOutputStream(fileOut);
            objectOut.writeObject(car);
            objectOut.close();
            fileOut.close();
            System.out.println("Serialized data is saved in vehicle.ser");
        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            FileInputStream fileIn = new FileInputStream("vehicle.ser");
            ObjectInputStream objectIn = new ObjectInputStream(fileIn);
            Vehicle car = (Vehicle) objectIn.readObject();
            objectIn.close();
            fileIn.close();
            System.out.println("\nDeserialized Vehicle:");
            System.out.println("Model: " + car.getModel());
            System.out.println("Engine Type: " + car.getEngine().getType());
        } catch (IOException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```



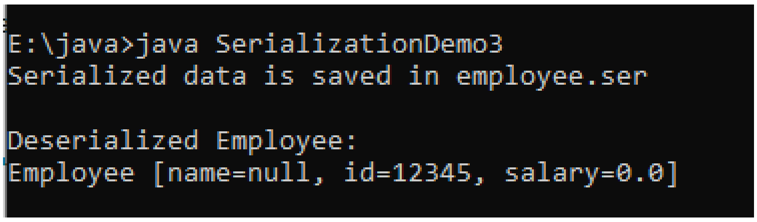
```
E:\java>java SerializationDemo2
Serialized data is saved in vehicle.ser

Deserialized Vehicle:
Model: Toyota
Exception in thread "main" java.lang.NullPointerException
        at SerializationDemo2.main(SerializationDemo2.java:71)
```

56. Write a program to Serialize/De-Serialize selected attributes of an Employee

```
import java.io.*;
class Employee implements Serializable {
    private static final long serialVersionUID = 1L;
    private transient String name;
    private int id;
    private transient double salary;
    public Employee(String name, int id, double salary) {
```

```
    this.name = name;
    this.id = id;
    this.salary = salary;
  }
  public String getName() {
    return name;
  }
  public int getId() {
    return id;
  }
  public double getSalary() {
    return salary;
  }
    private void writeObject(ObjectOutputStream out) throws IOException {
    out.defaultWriteObject();
    out.writeInt(id);
  }
  private void readObject(ObjectInputStream in) throws IOException,
ClassNotFoundException {
    in.defaultReadObject();
        id = in.readInt();
  }
  public String toString() {
    return "Employee [name=" + name + ", id=" + id + ", salary=" + salary + "]";
  }
}
public class SerializationDemo3 {
  public static void main(String[] args) {
       try {
      Employee employee = new Employee("John Doe", 12345, 50000);
      FileOutputStream fileOut = new FileOutputStream("employee.ser");
      ObjectOutputStream objectOut = new ObjectOutputStream(fileOut);
      objectOut.writeObject(employee);
```

```
            objectOut.close();
            fileOut.close();
            System.out.println("Serialized data is saved in employee.ser");
        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            FileInputStream fileIn = new FileInputStream("employee.ser");
            ObjectInputStream objectIn = new ObjectInputStream(fileIn);
            Employee employee = (Employee) objectIn.readObject();
            objectIn.close();
            fileIn.close();
            System.out.println("\nDeserialized Employee:");
            System.out.println(employee);
        } catch (IOException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

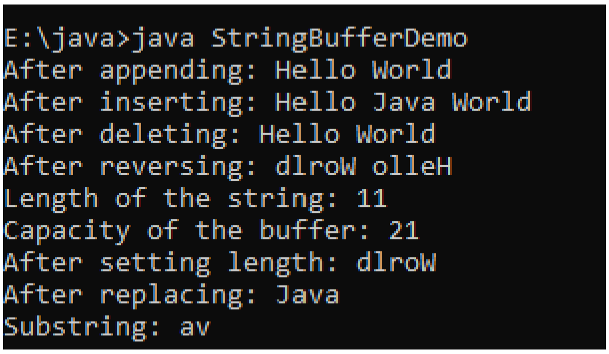


```
E:\java>java SerializationDemo3
Serialized data is saved in employee.ser

Deserialized Employee:
Employee [name=null, id=12345, salary=0.0]
```

57. Write a program to implement various methods of a StringBuffer class.

```
public class StringBufferDemo {
    public static void main(String[] args) {
        StringBuffer stringBuffer = new StringBuffer("Hello");
        stringBuffer.append(" World");
        System.out.println("After appending: " + stringBuffer);
        stringBuffer.insert(5, " Java");
       System.out.println("After inserting: " + stringBuffer);
```

```
        stringBuffer.delete(5, 10);
        System.out.println("After deleting: " + stringBuffer);
        stringBuffer.reverse();
        System.out.println("After reversing: " + stringBuffer);
        System.out.println("Length of the string: " + stringBuffer.length());
        System.out.println("Capacity of the buffer: " + stringBuffer.capacity());
        stringBuffer.setLength(5);
        System.out.println("After setting length: " + stringBuffer);
        stringBuffer.replace(0, 5, "Java");
        System.out.println("After replacing: " + stringBuffer);
        String subString = stringBuffer.substring(1, 3);
        System.out.println("Substring: " + subString);
    }
}
```

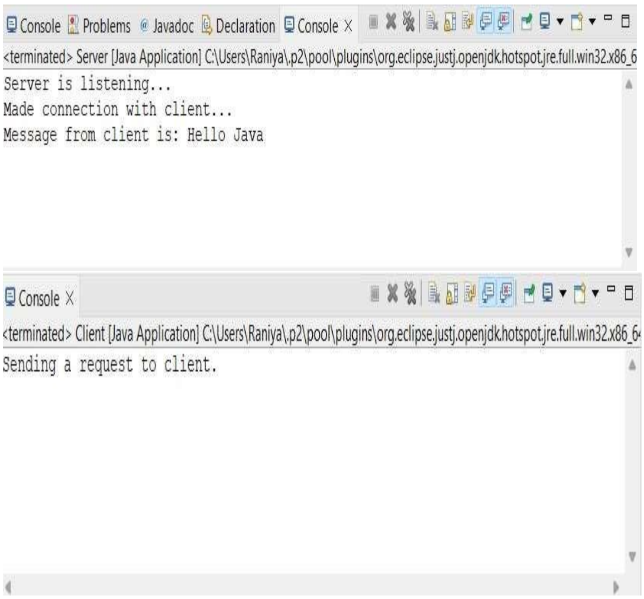


```
E:\java>java StringBufferDemo
After appending: Hello World
After inserting: Hello Java World
After deleting: Hello World
After reversing: dlroW olleH
Length of the string: 11
Capacity of the buffer: 21
After setting length: dlroW
After replacing: Java
Substring: av
```

58. Write a program to implement communication between a client and server client, via Socket Programming.

```
i package
clientserver; import
java.io.*; import
java.net.*; public
class Server

{
 public static void main(String[] args)
 {
    try
    {
            ServerSocket ss= new ServerSocket(12345);

            System.out.println("Server is listening...");

            Socket s=ss.accept();

            System.out.println("Made connection with client...");

            ObjectInputStream dis=new ObjectInputStream(s.getInputStream());

            String str=(String)dis.readUTF();

            System.out.println("Message from client is: "+str);

            ss.close();

            dis.close();

     }

     catch (IOException e)

     {

        e.printStackTrace();

}

}
 package clientserver;

 import java.io.*;

 import java.net.*;

 public class Client

 {

  public static void main(String[] args)

  {

     try

     {Socket s=new Socket("localhost",12345); System.out.println("Sending a request to client.");

            ObjectOutputStream dout=new ObjectOutputStream(s.getOutputStream());

            dout.writeUTF("Hello Java");

            dout.close();

     }

     catch (IOException e)

     {

        e.printStackTrace();

     }

   }
```

```
}
```



```
<terminated> Server [Java Application] C:\Users\Raniya\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_6
Server is listening...
Made connection with client...
Message from client is: Hello Java
```

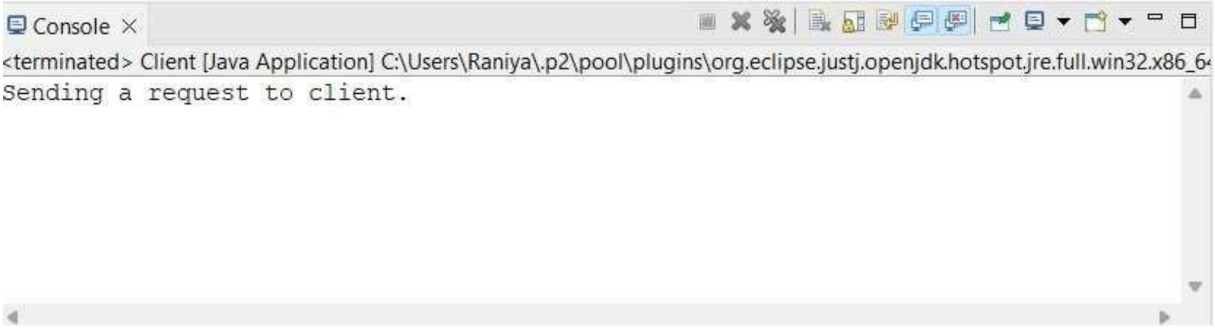
```
<terminated> Client [Java Application] C:\Users\Raniya\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
Sending a request to client.
```

SECTION 14

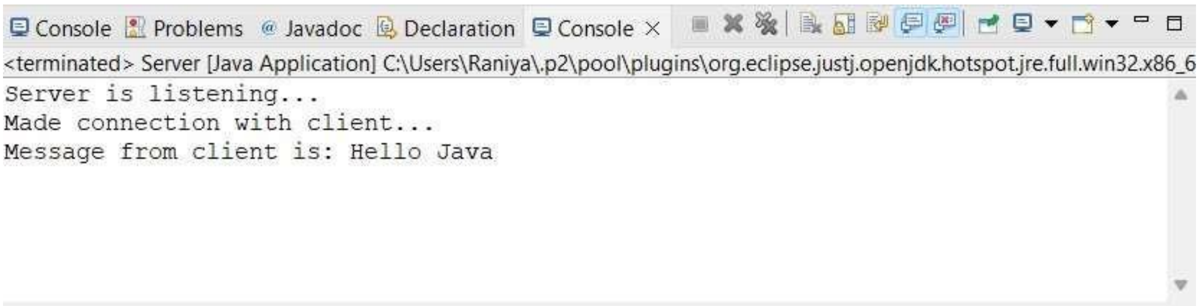
60. Accept a number from the client, and print its factorial in the server.

```
import java.io.*;

import java.net.*;


public class SockSer {

  public static void main(String[] args) {
    try {

      ServerSocket ss = new ServerSocket(12345);

      System.out.println("Waiting patiently for connection from client");


      Socket s = ss.accept(); // Wait for client to connect
      System.out.println("Made connection with client");


      BufferedReader reader = new BufferedReader(new InputStreamReader(s.getInputStream()));

      String input = reader.readLine(); // Read number from client


      int number = Integer.parseInt(input);

      long factorial = calculateFactorial(number);


      PrintWriter writer = new PrintWriter(s.getOutputStream(), true);
      writer.println(factorial); // Send factorial to client


      System.out.println("Factorial sent to client: " + factorial);

      reader.close();

      writer.close();

      s.close();

      s.close();

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

    }

  }


  private static long calculateFactorial(int number) {
    if (number == 0 || number == 1) {

      return 1;

    }
```

```
        long factorial = 1;

        for (int i = 2; i <= number; i++) {
            factorial *= i;

        }

        return factorial;

    }

}

import java.io.*;

import java.net.*;


public class SockCli {

    public static void main(String[] args) {
        try {

            Socket s = new Socket("localhost", 12345);
            System.out.println("Sending a request to server");

            BufferedReader userInputReader = new BufferedReader(new InputStreamReader(System.in));
            System.out.print("Enter a number to calculate its factorial: ");

            String userInput = userInputReader.readLine();

            PrintWriter writer = new PrintWriter(s.getOutputStream(), true);
            writer.println(userInput); // Send number to server

            BufferedReader reader = new BufferedReader(new InputStreamReader(s.getInputStream()));
            String response = reader.readLine(); // Receive factorial from server


            System.out.println("Factorial received from server: " + response);

            userInputReader.close();

            writer.close();

            reader.close();
            s.close();

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

        }

    }

}
```



```
<terminated> Client [Java Application] C:\Users\Raniya\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_6
Sending a request to client.
```



```
<terminated> Server [Java Application] C:\Users\Raniya\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_6
Server is listening...
Made connection with client...
Message from client is: Hello Java
```