JAVA: Practicals

1. Write a program to read two numbers from user and print their product.

(Introduction to Simple Java Program)

```java
import java.util.Scanner;
public class Simple{
 public static void main(String[] args) {
  // Create a Scanner object to read input from the user
  Scanner in = new Scanner(System.in);

  // Prompt the user to input the first number
  System.out.print("Input first number: ");
  // Read and store the first number
  int num1 = in.nextInt();

  // Prompt the user to input the second number
  System.out.print("Input second number: ");
  // Read and store the second number
    int num2 = in.nextInt();
  // Calculate the product of the two numbers and display the result
  System.out.println(num1 + " x " + num2 + " = " + num1 * num2);
 }
}
```

Output:

```
D:\java\classes\mypack>javac Simple.java

D:\java\classes\mypack>java Simple
Input first number: 50
Input second number: 40
50 x 40 = 2000
```

2. Write a program to print the square of a number passed through command line arguments.

(Command Line Arguments)

```
public class Simple{
 public static void main(String[] args) {
  // to read no from command line use args
         String num1=args[0];
         String num2=args[1];

  System.out.println("Square of "+num1 + " is " +
(Integer.parseInt(num1)*Integer.parseInt(num1)));
   System.out.println("Square of "+num2 + " is " +
(Integer.parseInt(num2)*Integer.parseInt(num2)));
 }
}
```

Output:

```
D:\java\classes\mypack>javac Simple.java

D:\java\classes\mypack>java Simple 4 6
Square of 4 is 16
Square of 6 is 36
```

3. Write a program to send the name and surname of a student through command line arguments and print a welcome message for the student.

(Command Line Arguments)

```
public class Simple{
 public static void main(String[] args) {
  // to read no from command line use args
          String fname=args[0];
          String lname=args[1];

  System.out.println("Hello, "+fname + " " +lname);

 }
}
```

Output:

```
D:\java\classes\mypack>javac Simple.java

D:\java\classes\mypack>java Simple Gopal Shinde
Hello, Gopal Shinde
```

4. Write a java program to find the largest number out of n natural numbers. (Relational operators)

```
import java.util.Scanner;
public class Simple{
 public static void main(String[] args) {
  int n, largest;
 Scanner s = new Scanner(System.in);
 System.out.println("Enter the size of the array:");
 n= s.nextInt();
 int arr[] =new int[n];
 System.out.println("Enter elements of array:");
 for(int i =0; i<n; i++)
 {
 arr[i] =s.nextInt( );
 }
 largest =arr[0];
 for(int i =0; i<n; i++)
 {
 if(largest < arr[i])
 {
 largest = arr[i];
 }
 }
 System.out.println("The largest number of " +n+ " natural numbers is: "+largest);
 }
}
```

Output:

```
Enter the size of the array:
5
Enter elements of array:
45
78
76
85
98
The largest number of 5 natural numbers is: 98
```

5. Write a java program to find the Fibonacci series & Factorial of a number using recursive and non-recursive functions. (Looping Statements)

Fibonacci series using recursive functions.

```
import java.io.*;
import java.util.Scanner;
public class Simple{

// Recursive method to calculate Fibonacci number
   public static int fibonacci(int n) {
      if (n <= 1) {
         return n; // Base case: return n for 0 and 1
      }
      return fibonacci(n - 1) + fibonacci(n - 2); // Recursive case
   }

 public static void main(String[] args) {

  int terms;
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter a number :");
 terms=sc.nextInt( );

      System.out.println("Fibonacci Series up to " + terms + " terms:");
      for (int i = 0; i < terms; i++) {
         System.out.println(fibonacci(i) + " ");
      }
 }
}
```

Output:

```
Enter a number :
5
Fibonacci Series up to 5 terms:
0 1 1 2 3
```

Fibonacci series Using non- recursive

```
import java.io.*;
import java.util.Scanner;
public class Simple{
 public static void main(String[] args) {
 int n,n1=0,n2=1,n3,i;
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter a number :");
 n=sc.nextInt( );
 System.out.println("The Fibonacci series is:");
 System.out.println(n1);
 System.out.println(n2);
 for(i=2; i<n ;i++)
 {
 n3 = n1+n2;
 System.out.println(n3);
 n1=n2;
 n2=n3;
 }
 }
}
```

Output:

```
Enter a number :
5
The Fibonacci series is:
0
1
1
2
3
```

Factorial using recursive functions

```
import java.io.*;
import java.util.Scanner;
public class Simple{

// Recursive method to calculate factorial
public static int factorial(int n) {
      // Base case: factorial of 0 or 1 is 1
      if (n == 0 || n == 1) {
         return 1;
      } else {
         // Recursive case: n * factorial(n - 1)
         return n * factorial(n - 1);
      }
   }


 public static void main(String[] args) {

         int number;
         Scanner sc=new Scanner(System.in);
         System.out.println("Enter a number :");
         number=sc.nextInt( );
         int result = factorial(number);
         System.out.println("Factorial of " + number + " is: " + result);

 }
}
```

Output:

```
Enter a number :
5
Factorial of 5 is: 120
```

Factorial using non-recursive

```
import java.io.*;
import java.util.Scanner;
public class Simple{

 public static void main(String[] args) {

          int n,i,fact=1;
 Scanner s=new Scanner(System.in);
 System.out.println("Enter a number :");
 n=s.nextInt( );
 for(i=1;i<=n;i++)
 {
 fact=fact*i;
 }
 System.out.println("The factorial of "+n+" is: "+fact);
 }
}
```

Output:

```
Enter a number :
5
The factorial of 5 is: 120
```

6. Write a java program to multiply two given matrices. (2D arrays)

```
import java.io.*;
import java.util.Scanner;
public class Simple{

 public static void main(String[] args) {

 int m, n, p, q, sum = 0, c, d, k;

Scanner in = new Scanner(System.in);
System.out.println("Enter the number of rows and columns of first matrix");
m = in.nextInt();
n = in.nextInt();

int first[][] = new int[m][n];

System.out.println("Enter elements of first matrix");

for (c = 0; c < m; c++)
for (d = 0; d < n; d++)
first[c][d] = in.nextInt();

System.out.println("Enter the number of rows and columns of second matrix");
p = in.nextInt();
q = in.nextInt();

if (n != p)
System.out.println("The matrices can't be multiplied with each other.");
else
{
int second[][] = new int[p][q];
int multiply[][] = new int[p][q];
System.out.println("Enter elements of second matrix");

for (c = 0; c < p; c++)
for (d = 0; d < q; d++)
second[c][d] = in.nextInt();

for (c = 0; c < m; c++) {
for (d = 0; d < q; d++) {
for (k = 0; k < p; k++)
sum = sum+first[c][k]*second[k][d];

multiply[c][d] = sum;
sum = 0;
}
}

System.out.println("Product of the matrices:");

for (c = 0; c < m; c++) {
for (d = 0; d < q; d++)
System.out.print(multiply[c][d]+"\t");

System.out.print("\n");

}
}
}
}
```

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

```
import java.io.*;
import java.util.Scanner;
public class Simple{

 public static void main(String[] args) {


         int n;
         Scanner sc=new Scanner(System.in);
         System.out.println("How many name you want to enter :");
         n=sc.nextInt( );

     Scanner s = new Scanner(System.in);
   String[] array = new String[n];

   System.out.println("Please enter "+n+" names to sort");

   for (int i = 0; i < array.length; i++) {
      array[i] = s.nextLine();
   }
String temp;
     for (int i = 0; i < array.length; i++) {
         for (int j = i + 1; j < array.length; j++) {

            // to compare one string with other strings
            if (array[i].compareTo(array[j]) > 0) {
               // swapping
               temp = array[i];
               array[i] = array[j];
               array[j] = temp;
            }
         }
```

```
        }

      // print output array
        System.out.println(
            "The names in alphabetical order are: ");
        for (int i = 0; i < array.length; i++) {
            System.out.println(array[i]);
        }

  }
}
```

Output:

```
How many name you want to enter :
5
Please enter 5 names to sort
Kiran
Shital
Ramesh
Pratik
Ajay
The names in alphabetical order are:
Ajay
Kiran
Pratik
Ramesh
Shital
```

8. Write a Java program that checks whether a given string is a palindrome or not . Ex:MADAM is a palindrome.

```
import java.io.*;
import java.util.Scanner;
public class Simple{

// Method to check if a string is a palindrome
  public static boolean isPalindrome(String s) {

     // Convert string to lowercase for
     // case-insensitive comparison
     s = s.toLowerCase();

     // Reverse the string
     String rev = "";
     for (int i = s.length() - 1; i >= 0; i--) {
       rev = rev + s.charAt(i);
     }

     // Compare the original string with
     // the reversed string
     return s.equals(rev);
  }


public static void main(String[] args) {



    // Input string
    Scanner s = new Scanner(System.in);
    System.out.println("Enter name to check palindrome or not :");
    String s1 = s.nextLine();
```

```
        // Check if the string is a palindrome
        boolean res = isPalindrome(s1);

        // Print the result
        if (res) {
           System.out.println("\"" + s1 + "\" is a palindrome.");
        } else {
           System.out.println("\"" + s1 + "\" is not a palindrome.");
        }

  }
}
```

Output:

```
Enter name to check palindrome or not :
LEVEL
"LEVEL" is a palindrome.
```

9. Write a java program to read n number of values in an array and display it in reverse order.

```
import java.io.*;
import java.util.Scanner;
import java.util.Arrays;
public class Simple{

public static void main(String[] args) {


        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("How many name you want to enter :");
        n=sc.nextInt();
        Scanner s = new Scanner(System.in);
        int[] arr = new int[n];

        System.out.println("Please enter "+n+" names to sort");

        for (int i = 0; i < arr.length; i++) {
        arr[i] = s.nextInt();
        }

     // Swap elements from start to end
     for (int i = 0; i < arr.length / 2; i++) {
        int t = arr[i];
        arr[i] = arr[arr.length - 1 - i];
        arr[arr.length - 1 - i] = t;
     }

     System.out.println("" + Arrays.toString(arr));

   }
}
```

Output:

```
How many name you want to enter:
5
Please enter 5 names to sort
40
30
20
15
10
[10, 15, 20, 30, 40]
```

10. Create a JAVA class called Student with the following details as variables within it.

a. USN, NAME, BRANCH, PHONE, PERCENTAGE

b. Write a JAVA program to create n Student objects and print the USN,Name, Branch, Phone,and percentage of these objects with suitable headings.

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

class Student
{
 String usn, name, branch;
 long ph;
double pr1;

 Student()
 {
 usn = name = branch = "no value";
 ph = 0;
pr1=0.0;
 }
 void read_data(String u, String n, String b, long p,double pr)
 {
 usn = u;
 name = n;
 branch = b;
 ph =p;
pr1=pr;
 }
void display()
 {
 System.out.println(usn + "\t\t" + name + "\t\t" + branch + "\t\t" + ph+"\t\t"+pr1);
 }
```

```
}

public class Simple{

public static void main(String[] args) {

        String u, n, b;
 long p;
 int no;
double per;
 Scanner br=new Scanner(System.in);
 System.out.println("Enter number of records");
 no = br.nextInt();

 Student[] s = new Student[no];

 for(int i=0; i<s.length;i++)
 {
 System.out.println("Enter " + (i + 1) + " Student record");
 s[i] = new Student();
 System.out.println("Enter student USN");
 u = br.next();
 System.out.println("Enter student Name");
 n = br.next();
 System.out.println("Enter student Branch");
 b = br.next();
 System.out.println("Enter student Phone number");
 p = br.nextLong();
System.out.println("Enter student Percentage");
 per = br.nextDouble();
 s[i].read_data(u, n, b, p,per);
 }

 System.out.println("USN    \t\t    NAME    \t\t    BRANCH    \t    PHONE
NO\tPERCENTAGE");
 for(int i=0; i<s.length;i++)
 {
 s[i].display();
 }
}
}
```

Output:

```
Enter number of records
1
Enter 1 Student record
Enter student USN
ABC0001
Enter student Name
GOpAL
Enter student Branch
LAtur
Enter student Phone number
9975349375
Enter student Percentage
45.55
USN         NAME        BRANCH        PHONE NO      PERCENTAGE
ABC0001       GOpAL         LAtur         9975349375          45.55
```

11. Write a Java program that displays the number of characters, lines and words in a text.

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

public class Simple{

public static void main(String[] args) {

try {
        Scanner input = new Scanner(new FileReader("data.txt"));
int lineCount = 0;
int wordCount = 0;
int charCount = 0;
while (input.hasNextLine()) {
  // read a line from the input file
  String line = input.nextLine();

  // increment line count
  lineCount++;

  // split line into words and increment word count
  String str [] = line.split((" "));
  for ( int i = 0; i <str.length ; i ++) {
    if (str [i].length() > 0) {
      wordCount ++;
    }
  }
  // increment char count
  charCount += (line.length());
}
      System.out.println("No. of Characters : "+charCount);
      System.out.println("No. of Lines : "+lineCount);
      System.out.println("No. of Words : "+wordCount);

      input.close();
 }

   catch (FileNotFoundException e) {
      System.out.println("There was an error opening one of the files.");
   }

   }
}
```

Output:

```
No. of Characters : 526
No. of Lines : 10
No. of Words : 100
```

12. Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea() methods to calculate the area and perimeter of a circle.

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

class Shape {

    // Public method to get the perimeter of the shape, returning a default value of 0.0
    public double getPerimeter() {
        return 0.0;
    }
    // Public method to get the area of the shape, returning a default value of 0.0
    public double getArea() {
        return 0.0;
    }
}

class Circle extends Shape {

    // Private instance variable for the radius of the circle
    private double radius;

    // Constructor for the Circle class, taking the radius as a parameter
    public Circle(double radius) {
        // Initialize the radius instance variable
        this.radius = radius;
    }

    // Override the getPerimeter method from the superclass (Shape)

 @Override
```

```
  public double getPerimeter() {
    // Return the perimeter of the circle calculated as 2 * π * radius
    return 2 * Math.PI * radius;
  }

  // Override the getArea method from the superclass (Shape)
@Override
  public double getArea() {
    // Return the area of the circle calculated as π * radius^2
    return Math.PI * radius * radius;
  }
}

public class Simple{

public static void main(String[] args) {

// Declare a double variable r and initialize it to 8.0
    double r = 10.0;

    // Create a Circle object named c1 with radius r
    Circle c1 = new Circle(r);

    // Print the radius of the circle c1
    System.out.println("Radius of the circle=" + r);

    // Print the perimeter of the circle c1
    System.out.println("Perimeter: " + c1.getPerimeter());

    // Print the area of the circle c1
    System.out.println("Area: " + c1.getArea());
  }
}
```

Output:

```
Radius of the circle=10.0
Perimeter: 62.83185307179586
Area: 314.1592653589793
```

13. Write a Java program to create a class Employee with a method called calculateSalary(). Create two subclasses Manager and Programmer. In each subclass, override the calculateSalary() method to calculate and return the salary based on their specific roles.

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

class Employee {

    // Declare private String variables name and role
    private String name;
    private String role;

    // Constructor for Employee class that takes name and role as parameters
    public Employee(String name, String role) {
        // Assign the parameter name to the instance variable name
        this.name = name;
        // Assign the parameter role to the instance variable role
        this.role = role;
    }

    // Public method to get the name of the employee
    public String getName() {
        // Return the name of the employee
        return name;
    }
    // Public method to get the role of the employee
    public String getRole() {
        // Return the role of the employee
        return role;
    }
    // Public method to calculate the salary of the employee
    public double calculateSalary() {
        // Return 0.0 as the default salary
        return 0.0;
    }
}

// Define the Manager class as a subclass of Employee
class Manager extends Employee {

    // Declare private double variables baseSalary and bonus
    private double baseSalary;
    private double bonus;

    // Constructor for Manager class that takes name, baseSalary, and bonus as parameters
    public Manager(String name, double baseSalary, double bonus) {
        // Call the constructor of the superclass Employee with name and role "Manager"
        super(name, "Manager");
        // Assign the parameter baseSalary to the instance variable baseSalary
        this.baseSalary = baseSalary;
        // Assign the parameter bonus to the instance variable bonus
        this.bonus = bonus;
    }

    // Override the calculateSalary method from the Employee class
    @Override
    public double calculateSalary() {
        // Calculate and return the salary of the manager by adding baseSalary and bonus
        return baseSalary + bonus;
    }
}
```

```
// Define the Programmer class as a subclass of Employee
class Programmer extends Employee {

   // Declare private double variables baseSalary and overtimePay
   private double baseSalary;
   private double overtimePay;

   // Constructor for Programmer class that takes name, baseSalary, and overtimePay as parameters
   public Programmer(String name, double baseSalary, double overtimePay) {
      // Call the constructor of the superclass Employee with name and role "Programmer"
      super(name, "Programmer");
      // Assign the parameter baseSalary to the instance variable baseSalary
      this.baseSalary = baseSalary;
      // Assign the parameter overtimePay to the instance variable overtimePay
      this.overtimePay = overtimePay;
   }

   // Override the calculateSalary method from the Employee class
   @Override
   public double calculateSalary() {
      // Calculate and return the salary of the programmer by adding baseSalary and overtimePay
      return baseSalary + overtimePay;
   }
}
```

```
public class Simple{

// Main method that serves as the entry point for the application
   public static void main(String[] args) {

        // Create a Manager object with name "Ramesh Patil", baseSalary 7500.0, and bonus 1500.0
        Employee emp1 = new Manager("Ramesh Patil", 7500.0, 1500.0);

        // Create a Programmer object with name "Kavya Shetty", baseSalary 5000.0, and overtimePay 600.0
        Employee emp2 = new Programmer("Kavya Shetty", 5000.0, 600.0);

        // Print the name, role, and salary of the Manager object
        System.out.println("Manager: " + emp1.getName() + "\nRole: " + emp1.getRole() + "\nSalary: " + emp1.calculateSalary());

        // Print the name, role, and salary of the Programmer object
        System.out.println("\nProgrammer: " + emp2.getName() + "\nRole: " + emp2.getRole() + "\nSalary: " + emp2.calculateSalary());

   }
}
```

Output:

```
Manager: Ramesh Patil
Role: Manager
Salary: 9000.0

Programmer: Kavya Shetty
Role: Programmer
Salary: 5600.0
```

14. Write a Java program using an interface called 'Bank' having function 'rate_of_interest()'.

Implement this interface to create two separate bank classes 'SBI' and 'PNB' to print different rates of interest. Include additional member variables, constructors also in classes 'SBI' and 'PNB'.

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



// Declare the Account interface
interface Bank {
   // Declare the abstract method "rate_of_interest()" to retrieve the interest rate
   double rate_of_interest();
}

// Declare the SavingsAccount class, which implements the Account interface
class SBI implements Bank {
    private double balance;
   private double interestRate;

   // Constructor for initializing the balance and interest rate
   public SBI (double initialDeposit,double interestRate) {
     this.balance = initialDeposit;
      this.interestRate = interestRate;
   }

  public void getBalance() {
System.out.println("Initial Deposit SBI : "+balance);
System.out.println("Interest Rate SBI: "+interestRate );

double ti=balance * interestRate / 100;
balance += balance * interestRate / 100;
System.out.println("Total Interest PNB: "+ti );
      System.out.println("Total Amount SBI : "+balance );
   }

   // Implement the "getBalance" method to retrieve the current balance
   @Override
   public double rate_of_interest() {
      return interestRate;
   }
}

// Declare the CurrentAccount class, which implements the Account interface
class PNB implements Bank {
   // Declare private instance variables to store balance and overdraft limit
   private double balance;
   private double pinterestRate;

   // Constructor for initializing the balance and overdraft limit
   public PNB(double initialDeposit,double pinterestRate){
      this.balance = initialDeposit;
      this.pinterestRate = pinterestRate;
   }
 public void getBalance() {

System.out.println("Initial Deposit PNB : "+balance);
System.out.println("Interest Rate PNB: "+pinterestRate );
double ti=balance * pinterestRate / 100;
balance += balance * pinterestRate / 100;
System.out.println("Total Interest PNB: "+ti );
      System.out.println("Total Amount PNB : "+balance );
   }



   // Implement the "getBalance" method to retrieve the current balance
```

```
  @Override
  public double rate_of_interest() {
    return pinterestRate;
  }
}

public class Simple{

  public static void main(String[] args) {

      SBI s=new SBI(1000,9.0);
      PNB p=new PNB(5000,7.0);
      System.out.println("SBI Interest Rate is : "+ s.rate_of_interest());
      s.getBalance();
      System.out.println("");
      System.out.println("PNB Interest Rate is : "+ p.rate_of_interest());
      p.getBalance();
  }
}
```

Output:

```
SBI Interest Rate is : 9.0
Initial Deposit SBI : 1000.0
Interest Rate SBI: 9.0
Total Interest PNB: 90.0
Total Amount SBI : 1090.0

PNB Interest Rate is : 7.0
Initial Deposit PNB : 5000.0
Interest Rate PNB: 7.0
Total Interest PNB: 350.0
Total Amount PNB : 5350.0
```

15. Write a Java package program for the class book and then import the

data from the package and display the result.

Bookinfo.java

```
package first;
public class Bookinfo
{
 int id;
 String name,author,pub,yy,price;
public Bookinfo(int bid,String bname,String bauthor,String bpub,String
byy,String bprice)
 {
  id=bid;
  name=bname;
  author=bauthor;
  pub=bpub;
  yy=byy;
  price=bprice;
 }
public void display()
 {
   System.out.println("BOOK DETAILS");
   System.out.println("BOOK ID : "+id);
   System.out.println("BOOK NAME : "+name);
   System.out.println("BOOK AUTHOR : "+author);
   System.out.println("BOOK PUBLISHER : "+pub);
   System.out.println("PUBLISH YEAR: "+yy);
   System.out.println("BOOK PRICE : "+price);
 }
}
```

PackageDemo.java

```
import first.*;

class PackageDemo

{

 public static void main(String args[])

 {

first.Bookinfo a=new first.Bookinfo(01,"OOPs using java","James Gosling","Sun microsystem","2025","1200");

  a.display();



}

}
```

Output:

```
BOOK DETAILS
BOOK ID : 1
BOOK NAME : OOPs using java
BOOK AUTHOR : James Gosling
BOOK PUBLISHER : Sun microsystem
PUBLISH YEAR: 2025
BOOK PRICE : 1200
```

16. Write a Java program for finding the cube of a number using a package

for various data types and then import it in another class and display the results.

Cube.java

```
package cube;
public class Cube
{
        public static long getCube(long x)
        {
        return x*x*x;
        }
}
```

CubeMain.java

```
class CubeMain
{
        public static void main(String args[])
        {
        long l;
        Scanner c=new Scanner(System.in);
        System.out.println("Enter Number to find Cube ");
        l=c.nextLong();
        System.out.println("Cube is : "+Cube.getCube(l));
        }
}
```

Output:

```
Enter Number to find Cube
1500
Cube is : 3375000000
```

17. Write a Java program for demonstrating the divide by zero exception

handling.

```
import java.util.*;

public class Simple{

   public static void main(String[] args) {

        Scanner c=new Scanner(System.in);
        System.out.println("Enter Numerator Value");

        int a = c.nextInt();
        System.out.println("Enter Denominator Value");
       int b = c.nextInt();
       try {
          System.out.println("Division is : "+ (double)a / (double)b); // throw Exception
       }
       catch (ArithmeticException e) {
          // Exception handler
          System.out.println(
             "Divided by zero operation cannot possible");
       }

   }
}
```

Output:

```
Enter Numerator Value
12
Enter Denominator Value
5
Division is : 2.4
```

18. Write a Java program that reads a list of integers from the user and

throws an exception if any numbers are duplicates.

```
import java.util.*;

public class Simple{

  public static void main(String[] args) {

try {
   List < Integer > numbers = readNumbersFromUser();
   checkDuplicates(numbers);
   System.out.println("No duplicate numbers!");
  } catch (Duplicate_Number_Exception e) {
   System.out.println("Error: " + e.getMessage());
  }
 }

 public static List < Integer > readNumbersFromUser() {
  List < Integer > numbers = new ArrayList < > ();
  Scanner scanner = new Scanner(System.in);

  System.out.print("How many numbers do you want to input? ");
  int count = scanner.nextInt();

  System.out.println("Input the integers:");
  for (int i = 0; i < count; i++) {
   int num = scanner.nextInt();
   numbers.add(num);
  }

  scanner.close();
  return numbers;
 }
 public static void checkDuplicates(List < Integer > numbers) throws
Duplicate_Number_Exception {
  Set < Integer > uniqueNumbers = new HashSet < > ();

  for (int num: numbers) {
    if (uniqueNumbers.contains(num)) {
      throw new Duplicate_Number_Exception("Duplicate number found: " + num);
    }
    uniqueNumbers.add(num);
  }
  }
}
class Duplicate_Number_Exception extends Exception {
 public Duplicate_Number_Exception(String message) {
  super(message);
 }
}
```

Output:

```
How many numbers do you want to input? 5
Input the integers:
10
20
10
30
20
Error: Duplicate number found: 10
```

19. Create an exception subclass UnderAge, which prints “Under Age” along

with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws UnderAge exception if the variable age passed to it as argument is less than 18.

Write main() method also to show working of the program.

```
import java.util.*;



class UnderAge extends Exception {

 final private int age;

 public UnderAge(int age) {

 this.age = age;

 }

 @Override

 public String getMessage() {

 return "UnderAge: " + age + " is less than 18";

 }

}



public class Simple{

static void test(int age) throws UnderAge {

 if (age < 18)
 throw new UnderAge(age);
 }
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
 System.out.print("Enter Age: ");
 int age = sc.nextInt();
 try {
 test(age);
 System.out.println("Test Successful");
 } catch (UnderAge e) {
 System.err.println(e.getMessage());
 System.out.println("Test Unsuccessful");
 } finally {
 sc.close();
 }
}
}
```

Output:

```
Enter Age: 18

Test Successful



Enter Age: 15

UnderAge: 15 is less than 18

Test Unsuccessful
```

Thank You!