SUBJECT: CA 511 MJP Lab Course Based on CA 510A MJ (Java Programming)

Assignment 1

Q.1) Write a Java program to accept a number from user and generate multiplication table of a number. Accept number using BufferedReader class. [10 Marks]

```
import java.io.*;

public class MultiplicationTable {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter a number: ");
        int num = Integer.parseInt(br.readLine());
        System.out.println("Multiplication table of " + num + ":");
        for (int i = 1; i <= 10; i++) {
            System.out.println(num + " x " + i + " = " + (num * i));
        }
    }
}
```

Output:

```
Enter a number: 5
Multiplication table of 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Q.2) Define a class MyNumber having one private integer data member. Write a default constructor to initialize it to 0 and another constructor to initialize it to a value. Write methods `isNegative`, `isPositive`, `isOdd`, `isEven`. Use command line argument to pass a value to the object and perform the above operations. [20 Marks]

```
class MyNumber {
    private int num;

    MyNumber() {
        num = 0;
    }
    MyNumber(int n) {
        num = n;
    }

    boolean isPositive() {
        return num > 0;
    }

    boolean isNegative() {
        return num < 0;
    }

    boolean isEven() {
        return num % 2 == 0;
    }

    boolean isOdd() {
        return num % 2 != 0;
    }
}

public class TestMyNumber {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("Please provide a number as command line
argument.");
            return;
        }

        int value = Integer.parseInt(args[0]);
        MyNumber obj = new MyNumber(value);

        System.out.println("Number: " + value);
        System.out.println("Is Positive? " + obj.isPositive());
        System.out.println("Is Negative? " + obj.isNegative());
        System.out.println("Is Even? " + obj.isEven());
        System.out.println("Is Odd? " + obj.isOdd());
    }
}
```

Output (command line argument 7):

```
Number: 7
Is Positive? true
Is Negative? false
Is Even? false
Is Odd? true
```

Assignment 2

Q.1) Write a Java Program to Reverse a Number. Accept number using command line arguments. [10 Marks]

```
public class ReverseNumber {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("Please provide a number as command line
argument.");
            return;
        }

        int num = Integer.parseInt(args[0]);
        int reversed = 0;

        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }

        System.out.println("Reversed Number: " + reversed);
    }
}
```

Output (command line argument 12345):

```
Reversed Number: 54321
```

Q.2) Write a program to create class Account (accno, accname, balance). Create an array of 'n' Account objects. Define static method “sortAccount” which sorts the array on the basis of balance. Display account details in sorted order. [20 Marks]

```
class Account {
    int accno;
    String accname;
    double balance;

    Account(int accno, String accname, double balance) {
        this.accno = accno;
        this.accname = accname;
        this.balance = balance;
    }

    static void sortAccount(Account[] arr) {
        for (int i = 0; i < arr.length - 1; i++) {
            for (int j = 0; j < arr.length - i - 1; j++) {
                if (arr[j].balance > arr[j + 1].balance) {
                    Account temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }

    void display() {
        System.out.println(accno + "\t" + accname + "\t" + balance);
    }
}

public class TestAccount {
    public static void main(String[] args) {
        Account[] accounts = new Account[3];
        accounts[0] = new Account(101, "shivraj", 5000);
        accounts[1] = new Account(102, "ajay", 3000);
        accounts[2] = new Account(103, "swapnil", 7000);

        Account.sortAccount(accounts);

        System.out.println("Account Details in Sorted Order (by balance):");
        System.out.println("AccNo\tAccName\tBalance");
        for (Account a : accounts) {
            a.display();
        }
    }
}
```

Output:

```
Account Details in Sorted Order (by balance):
AccNo   AccName Balance
102     ajay     3000.0
101     shivraj   5000.0
103     swapnil 7000.0
```

Assignment 3

Q.1) Write a Java program to print the sum of elements of the array. Also display array elements in ascending order. [10 Marks]

```
import java.util.Arrays;

public class ArraySum {
    public static void main(String[] args) {
        int[] arr = {10, 5, 8, 12, 3};
        int sum = 0;

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

        Arrays.sort(arr);

        System.out.print("Array in ascending order: ");
        for (int num : arr) {
            System.out.print(num + " ");
        }

        System.out.println("\nSum of array elements: " + sum);
    }
}
```

Output:

```
Array in ascending order: 3 5 8 10 12
Sum of array elements: 38
```

Q.2) Write a program which defines class Product with data members id, name, and price. Store the information of 5 products and display the name of product having minimum price (Use array of objects). [20 Marks]

```
class Product {
    int id;
    String name;
    double price;

    Product(int id, String name, double price) {
        this.id = id;
        this.name = name;
        this.price = price;
    }
}

public class TestProduct {
    public static void main(String[] args) {
        Product[] products = new Product[5];
        products[0] = new Product(101, "Laptop", 55000);
        products[1] = new Product(102, "Mouse", 500);
        products[2] = new Product(103, "Keyboard", 1200);
        products[3] = new Product(104, "Monitor", 7000);
        products[4] = new Product(105, "USB Drive", 300);

        Product minProduct = products[0];
        for (int i = 1; i < products.length; i++) {
            if (products[i].price < minProduct.price) {
                minProduct = products[i];
            }
        }

        System.out.println("Product with minimum price: " + minProduct.name);
    }
}
```

Output:

```
Product with minimum price: USB Drive
```

Assignment 4

Q.1) Write a Java program to print the factors of a given number. (Use Scanner class). [10 Marks]

```
import java.util.Scanner;

public class Factors {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();

        System.out.print("Factors of " + num + " are: ");
        for (int i = 1; i <= num; i++) {
            if (num % i == 0) {
                System.out.print(i + " ");
            }
        }
        sc.close();
    }
}
```

Sample Output:

```
Enter a number: 12
Factors of 12 are: 1 2 3 4 6 12
```

Q.2) Construct a linked list containing names of colours: red, blue, yellow, and orange. Then extend the program to do the following:

i. Display the contents of the List using an Iterator
ii. Display the contents of the List in reverse order using a ListIterator
iii. Create another list containing pink and green. Insert the elements of this list between blue and yellow. [20 Marks]

```
import java.util.*;

public class ColorList {
    public static void main(String[] args) {
        LinkedList<String> colors = new LinkedList<>();
        colors.add("red");
        colors.add("blue");
        colors.add("yellow");
        colors.add("orange");

        System.out.println("Original List using Iterator:");
        Iterator<String> it = colors.iterator();
        while (it.hasNext()) {
            System.out.print(it.next() + " ");
        }

        System.out.println("\nList in reverse order using ListIterator:");
        ListIterator<String> listIt = colors.listIterator(colors.size());
        while (listIt.hasPrevious()) {
            System.out.print(listIt.previous() + " ");
        }

        LinkedList<String> newColors = new LinkedList<>();
        newColors.add("pink");
        newColors.add("green");

        int index = colors.indexOf("yellow");
        colors.addAll(index, newColors);

        System.out.println("\nUpdated List after inserting pink and green:");
        for (String color : colors) {
            System.out.print(color + " ");
        }
    }
}
```

Sample Output:

```
Original List using Iterator:
red blue yellow orange
List in reverse order using ListIterator:
orange yellow blue red
Updated List after inserting pink and green:
red blue pink green yellow orange
```

Assignment 5

Q.1) Write a Java program to accept a number from user and print all prime numbers up to that number (Use BufferedReader class). [10 Marks]

```
import java.io.*;

public class PrimeNumbers {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter a number: ");
        int n = Integer.parseInt(br.readLine());

        System.out.println("Prime numbers up to " + n + ":");
        for (int i = 2; i <= n; i++) {
            boolean isPrime = true;
            for (int j = 2; j <= Math.sqrt(i); j++) {
                if (i % j == 0) {
                    isPrime = false;
                    break;
                }
            }
            if (isPrime) {
                System.out.print(i + " ");
            }
        }
    }
}
```

Sample Output:

```
Enter a number: 20
Prime numbers up to 20:
2 3 5 7 11 13 17 19
```

Q.2) Create a package “utility”. Define a class `CapitalString` under “utility” package which will contain a method to return String with first letter capital. Create a `Person` class (members – name, city) outside the package. Display the person’s name with first letter as capital by making use of `CapitalString`. [20 Marks]

File 1: `CapitalString.java` (inside `utility` package)

```
package utility;

public class CapitalString {
    public static String capitalize(String str) {
        if (str == null || str.isEmpty()) {
            return str;
        }
        return str.substring(0, 1).toUpperCase() + str.substring(1);
    }
}
```

File 2: `Person.java` (outside package)

```
import utility.CapitalString;

public class Person {
    String name;
    String city;

    Person(String name, String city) {
        this.name = name;
        this.city = city;
    }

    void display() {
        System.out.println("Name: " + CapitalString.capitalize(name));
        System.out.println("City: " + city);
    }

    public static void main(String[] args) {
        Person p = new Person("john", "pune");
        p.display();
    }
}
```

Sample Output:

```
Name: John
City: pune
```

Assignment 6

Q.1) Write a Java Program to Display Armstrong Numbers Between Range. Accept range from user. [10 Marks]

```
import java.util.Scanner;

public class ArmstrongRange {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter start of range: ");
        int start = sc.nextInt();
        System.out.print("Enter end of range: ");
        int end = sc.nextInt();

        System.out.println("Armstrong numbers between " + start + " and " + end +
":");
        for (int i = start; i <= end; i++) {
            int num = i;
            int sum = 0;
            int digits = String.valueOf(i).length();

            while (num != 0) {
                int digit = num % 10;
                sum += Math.pow(digit, digits);
                num /= 10;
            }

            if (sum == i) {
                System.out.print(i + " ");
            }
        }
        sc.close();
    }
}
```

Sample Output:

```
Enter start of range: 100
Enter end of range: 500
Armstrong numbers between 100 and 500:
153 370 371 407
```

Q.2) Write a program to create class `MyDate` with `dd, mm, yy` as data members. Write parameterized constructor. Display the date in dd-mm-yy format. (Use `this` keyword). [20 Marks]

```
class MyDate {
    int dd, mm, yy;

    MyDate(int dd, int mm, int yy) {
        this.dd = dd;
        this.mm = mm;
        this.yy = yy;
    }

    void display() {
        System.out.println(dd + "-" + mm + "-" + yy);
    }

    public static void main(String[] args) {
        MyDate date = new MyDate(13, 11, 2025);
        date.display();
    }
}
```

Sample Output:

```
13-11-2025
```

Assignment 7

Q.1) Write Java program to check whether number is Perfect or not. [10 Marks]

```
import java.util.Scanner;

public class PerfectNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        int sum = 0;

        for (int i = 1; i <= num / 2; i++) {
            if (num % i == 0) {
                sum += i;
            }
        }

        if (sum == num) {
            System.out.println(num + " is a Perfect number.");
        } else {
            System.out.println(num + " is not a Perfect number.");
        }
        sc.close();
    }
}
```

Sample Output 1:

```
Enter a number: 28
28 is a Perfect number.
```

Sample Output 2:

```
Enter a number: 15
15 is not a Perfect number.
```

Q.2) Define a class Student with attributes rollno and name. Define default and parameterized constructor. Keep the count of objects created. Create objects using parameterized constructor and display the object count after each object is created. [20 Marks]

```
class Student {
    int rollno;
    String name;
    static int count = 0;

    Student() {
        count++;
    }

    Student(int rollno, String name) {
        this.rollno = rollno;
        this.name = name;
        count++;
        System.out.println("Object created. Total objects: " + count);
    }

    void display() {
        System.out.println("Roll No: " + rollno + ", Name: " + name);
    }

    public static void main(String[] args) {
        Student s1 = new Student(101, "Shivraj");
        Student s2 = new Student(102, "Ajay");
        Student s3 = new Student(103, "Swapnil");
    }
}
```

Sample Output:

```
Object created. Total objects: 1
Object created. Total objects: 2
Object created. Total objects: 3
```

Assignment 8

Q.1) Define a class `Student` having rollno, name, and percentage. Define Default and parameterized constructor. Accept the 5 student details and display it. (Use `this` keyword). [10 Marks]

```
import java.util.Scanner;

class Student {
    int rollno;
    String name;
    double percentage;

    Student() {
        this.rollno = 0;
        this.name = "";
        this.percentage = 0.0;
    }

    Student(int rollno, String name, double percentage) {
        this.rollno = rollno;
        this.name = name;
        this.percentage = percentage;
    }

    void display() {
        System.out.println("Roll No: " + this.rollno + ", Name: " + this.name + ", 
Percentage: " + this.percentage);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Student[] students = new Student[5];

        for (int i = 0; i < 5; i++) {
            System.out.print("Enter roll no: ");
            int r = sc.nextInt();
            sc.nextLine(); // consume newline
            System.out.print("Enter name: ");
            String n = sc.nextLine();
            System.out.print("Enter percentage: ");
            double p = sc.nextDouble();
            students[i] = new Student(r, n, p);
        }

        System.out.println("\nStudent Details:");
        for (Student s : students) {
            s.display();
        }

        sc.close();
    }
}
```

Sample Output:

```
Enter roll no: 101
Enter name: Shivraj
Enter percentage: 85
Enter roll no: 102
Enter name: Ajay
Enter percentage: 78
Enter roll no: 103
Enter name: Swapnil
Enter percentage: 92
Enter roll no: 104
Enter name: David
Enter percentage: 88
Enter roll no: 105
Enter name: Eva
Enter percentage: 80

Student Details:
Roll No: 101, Name: Shivraj, Percentage: 85.0
Roll No: 102, Name: Ajay, Percentage: 78.0
Roll No: 103, Name: Swapnil, Percentage: 92.0
Roll No: 104, Name: David, Percentage: 88.0
Roll No: 105, Name: Eva, Percentage: 80.0
```

Q.2) Create an abstract class `Shape` with methods `area` & `volume`. Derive a class `Cylinder` (radius, height). Calculate area and volume. [20 Marks]

```
abstract class Shape {
    abstract double area();
    abstract double volume();
}

class Cylinder extends Shape {
    double radius, height;

    Cylinder(double radius, double height) {
        this.radius = radius;
        this.height = height;
    }

    double area() {
        return 2 * Math.PI * radius * (radius + height);
    }

    double volume() {
        return Math.PI * radius * radius * height;
    }

    public static void main(String[] args) {
        Cylinder c = new Cylinder(5, 10);
        System.out.println("Area of Cylinder: " + c.area());
        System.out.println("Volume of Cylinder: " + c.volume());
    }
}
```

Sample Output:

```
Area of Cylinder: 471.23889803846896
Volume of Cylinder: 785.3981633974483
```

Assignment 9

Q.1) Write a Java program to display Fibonacci series using function. [10 Marks]

```
import java.util.Scanner;

public class FibonacciSeries {
    static void displayFibonacci(int n) {
        int a = 0, b = 1;
        System.out.print("Fibonacci Series: ");
        for (int i = 1; i <= n; i++) {
            System.out.print(a + " ");
            int next = a + b;
            a = b;
            b = next;
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of terms: ");
        int n = sc.nextInt();
        displayFibonacci(n);
        sc.close();
    }
}
```

Sample Output:

```
Enter number of terms: 8
Fibonacci Series: 0 1 1 2 3 5 8 13
```

Q.2) Define an interface `Operation` which has methods `area()` and `volume()`. Define a constant `PI` having a value 3.142. Create a class `Circle` (member – radius), `Cylinder` (members – radius, height) which implements this interface. Calculate and display the area and volume. [20 Marks]

```
interface Operation {
    double PI = 3.142;
    double area();
    double volume();
}

class Circle implements Operation {
    double radius;

    Circle(double radius) {
        this.radius = radius;
    }

    public double area() {
```

```java
        return PI * radius * radius;
    }

    public double volume() {
        return 0;
    }
}

class Cylinder implements Operation {
    double radius, height;

    Cylinder(double radius, double height) {
        this.radius = radius;
        this.height = height;
    }

    public double area() {
        return 2 * PI * radius * (radius + height);
    }

    public double volume() {
        return PI * radius * radius * height;
    }
}

public class TestOperation {
    public static void main(String[] args) {
        Circle c = new Circle(5);
        Cylinder cy = new Cylinder(5, 10);

        System.out.println("Circle Area: " + c.area());
        System.out.println("Circle Volume: " + c.volume());
        System.out.println("Cylinder Area: " + cy.area());
        System.out.println("Cylinder Volume: " + cy.volume());
    }
}
```

Sample Output:

```
Circle Area: 78.55
Circle Volume: 0.0
Cylinder Area: 471.3
Cylinder Volume: 785.5
```

Assignment 10

Q.1) Write a program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, and the type of file. [10 Marks]

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

public class FileInfo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter file name with path: ");
        String filename = sc.nextLine();

        File file = new File(filename);

        System.out.println("File Exists: " + file.exists());
        System.out.println("Readable: " + file.canRead());
        System.out.println("Writable: " + file.canWrite());
        if (file.exists()) {
            if (file.isFile()) {
                System.out.println("Type: Regular File");
            } else if (file.isDirectory()) {
                System.out.println("Type: Directory");
            } else {
                System.out.println("Type: Unknown");
            }
        }

        sc.close();
    }
}
```

Sample Output:

```
Enter file name with path: test.txt
File Exists: true
Readable: true
Writable: true
Type: Regular File
```

Q.2) Write a Java program to create a superclass `Vehicle` having members `Company` and `Price`. Derive two different classes `LightMotorVehicle` (mileage) and `HeavyMotorVehicle` (capacity_in_tons). Accept the information for "n" vehicles and display the information in appropriate form. While taking data, ask the user about the type of vehicle first. [20 Marks]

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

```
class Vehicle {
    String company;
    double price;

    Vehicle(String company, double price) {
        this.company = company;
        this.price = price;
    }

    void display() {
        System.out.println("Company: " + company + ", Price: " + price);
    }
}

class LightMotorVehicle extends Vehicle {
    double mileage;

    LightMotorVehicle(String company, double price, double mileage) {
        super(company, price);
        this.mileage = mileage;
    }

    void display() {
        super.display();
        System.out.println("Mileage: " + mileage);
    }
}

class HeavyMotorVehicle extends Vehicle {
    double capacityInTons;

    HeavyMotorVehicle(String company, double price, double capacityInTons) {
        super(company, price);
        this.capacityInTons = capacityInTons;
    }

    void display() {
        super.display();
        System.out.println("Capacity in Tons: " + capacityInTons);
    }
}

public class TestVehicle {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of vehicles: ");
        int n = sc.nextInt();
        sc.nextLine();

        Vehicle[] vehicles = new Vehicle[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter type of vehicle (light/heavy): ");
            String type = sc.nextLine().toLowerCase();
            System.out.print("Enter company name: ");
            String company = sc.nextLine();
            System.out.print("Enter price: ");
            double price = sc.nextDouble();

            if (type.equals("light")) {
                System.out.print("Enter mileage: ");
                double mileage = sc.nextDouble();
                sc.nextLine();
                vehicles[i] = new LightMotorVehicle(company, price, mileage);
            } else if (type.equals("heavy")) {
                System.out.print("Enter capacity in tons: ");
                double capacity = sc.nextDouble();
                sc.nextLine();
                vehicles[i] = new HeavyMotorVehicle(company, price, capacity);
            } else {
                System.out.println("Invalid type! Skipping this vehicle.");
                i--;
            }
        }

        System.out.println("\nVehicle Details:");
        for (Vehicle v : vehicles) {
            v.display();
            System.out.println("-------------------");
        }

        sc.close();
    }
}
```

Sample Output:

```
Enter number of vehicles: 2
Enter type of vehicle (light/heavy): light
Enter company name: Honda
Enter price: 500000
Enter mileage: 25
Enter type of vehicle (light/heavy): heavy
Enter company name: Tata
Enter price: 2000000
Enter capacity in tons: 10

Vehicle Details:
Company: Honda, Price: 500000.0
Mileage: 25.0
-------------------
Company: Tata, Price: 2000000.0
Capacity in Tons: 10.0
-------------------
```

Assignment 11

Q.1) Write a Java program to accept a number from the user. If number is zero then throw user-defined exception “Number is 0”, otherwise check whether the number is prime or not. [10 Marks]

```
import java.util.Scanner;

class ZeroNumberException extends Exception {
    ZeroNumberException(String msg) {
        super(msg);
    }
}

public class PrimeCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();

        try {
            if (num == 0) {
                throw new ZeroNumberException("Number is 0");
            }

            boolean isPrime = true;
            if (num == 1) {
                isPrime = false;
            } else {
                for (int i = 2; i <= Math.sqrt(num); i++) {
                    if (num % i == 0) {
                        isPrime = false;
                        break;
                    }
                }
            }

            if (isPrime) {
                System.out.println(num + " is a Prime number.");
            } else {
                System.out.println(num + " is not a Prime number.");
            }

        } catch (ZeroNumberException e) {
            System.out.println(e.getMessage());
        }

        sc.close();
    }
}
```

Sample Output 1:

```
Enter a number: 0
Number is 0
```

Sample Output 2:

```
Enter a number: 7
7 is a Prime number.
```

Sample Output 3:

```
Enter a number: 12
12 is not a Prime number.
```

Q.2) Create an abstract class `Shape` with methods `area` & `volume`. Derive a class `Cylinder` (radius, height). Calculate area and volume. [20 Marks]

```
abstract class Shape {
    abstract double area();
    abstract double volume();
}

class Cylinder extends Shape {
    double radius, height;

    Cylinder(double radius, double height) {
        this.radius = radius;
        this.height = height;
    }

    double area() {
        return 2 * Math.PI * radius * (radius + height);
    }

    double volume() {
        return Math.PI * radius * radius * height;
    }

    public static void main(String[] args) {
        Cylinder c = new Cylinder(5, 10);
        System.out.println("Area of Cylinder: " + c.area());
        System.out.println("Volume of Cylinder: " + c.volume());
    }
}
```

Sample Output:

```
Area of Cylinder: 471.23889803846896
Volume of Cylinder: 785.3981633974483
```

Assignment 12

Q.1) Write a Java Program to Display Armstrong Numbers Between Range. Accept range from user. [10 Marks]

```
import java.util.Scanner;

public class ArmstrongRange {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter start of range: ");
        int start = sc.nextInt();
        System.out.print("Enter end of range: ");
        int end = sc.nextInt();

        System.out.println("Armstrong numbers between " + start + " and " + end +
":");
        for (int i = start; i <= end; i++) {
            int num = i;
            int sum = 0;
            int digits = String.valueOf(i).length();

            while (num != 0) {
                int digit = num % 10;
                sum += Math.pow(digit, digits);
                num /= 10;
            }

            if (sum == i) {
                System.out.print(i + " ");
            }
        }
        sc.close();
    }
}
```

Sample Output:

```
Enter start of range: 100
Enter end of range: 500
Armstrong numbers between 100 and 500:
153 370 371 407
```

Q.2) Define an Interface `Shape` with abstract method `area()`. Write a Java program to calculate the area of Circle and Sphere. (Use `final` keyword). [20 Marks]

```
interface Shape {
    double area();
}

final class Circle implements Shape {
```

```java
    double radius;

    Circle(double radius) {
        this.radius = radius;
    }

    public double area() {
        return Math.PI * radius * radius;
    }
}

final class Sphere implements Shape {
    double radius;

    Sphere(double radius) {
        this.radius = radius;
    }

    public double area() {
        return 4 * Math.PI * radius * radius;
    }
}

public class TestShape {
    public static void main(String[] args) {
        Circle c = new Circle(5);
        Sphere s = new Sphere(5);

        System.out.println("Area of Circle: " + c.area());
        System.out.println("Area of Sphere: " + s.area());
    }
}
```

Sample Output:

```
Area of Circle: 78.53981633974483
Area of Sphere: 314.1592653589793
```

Assignment 13

Q.1) Write a Java program to count number of digits, spaces, and characters from a file. [10 Marks]

```
import java.io.*;

public class CountFile {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter file name with path: ");
        String filename = br.readLine();

        File file = new File(filename);
        if (!file.exists()) {
            System.out.println("File does not exist.");
            return;
        }

        FileReader fr = new FileReader(file);
        int ch;
        int digits = 0, spaces = 0, characters = 0;

        while ((ch = fr.read()) != -1) {
            char c = (char) ch;
            if (Character.isDigit(c))
                digits++;
            else if (Character.isWhitespace(c))
                spaces++;
            else
                characters++;
        }

        System.out.println("Digits: " + digits);
        System.out.println("Spaces: " + spaces);
        System.out.println("Characters: " + characters);

        fr.close();
    }
}
```

Sample Output (assuming file contains "Hello 123 World"):

```
Digits: 3
Spaces: 2
Characters: 10
```

Q.2) Create a package named “Series” having three different classes to print series:

a. Fibonacci series
b. Cube of numbers

c. Square of numbers
Write a Java program to generate “n” terms of the above series. Accept n from user. [20 Marks]

File 1: Fibonacci.java

```
package Series;

public class Fibonacci {
    public void printSeries(int n) {
        int a = 0, b = 1;
        System.out.print("Fibonacci Series: ");
        for (int i = 1; i <= n; i++) {
            System.out.print(a + " ");
            int next = a + b;
            a = b;
            b = next;
        }
        System.out.println();
    }
}
```

File 2: CubeSeries.java

```
package Series;

public class CubeSeries {
    public void printSeries(int n) {
        System.out.print("Cube Series: ");
        for (int i = 1; i <= n; i++) {
            System.out.print(i * i * i + " ");
        }
        System.out.println();
    }
}
```

File 3: SquareSeries.java

```
package Series;

public class SquareSeries {
    public void printSeries(int n) {
        System.out.print("Square Series: ");
        for (int i = 1; i <= n; i++) {
            System.out.print(i * i + " ");
        }
        System.out.println();
    }
}
```

File 4: TestSeries.java

```
import java.util.Scanner;
import Series.Fibonacci;
import Series.CubeSeries;
import Series.SquareSeries;

public class TestSeries {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of terms: ");
        int n = sc.nextInt();

        Fibonacci f = new Fibonacci();
        CubeSeries c = new CubeSeries();
        SquareSeries s = new SquareSeries();

        f.printSeries(n);
        c.printSeries(n);
        s.printSeries(n);

        sc.close();
    }
}
```

Sample Output (n = 5):

```
Enter number of terms: 5
Fibonacci Series: 0 1 1 2 3
Cube Series: 1 8 27 64 125
Square Series: 1 4 9 16 25
```

Assignment 14

Q.1) Write a program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, and the type of file. [10 Marks]

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

public class FileInformation {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter file name with path: ");
        String filename = sc.nextLine();

        File file = new File(filename);

        System.out.println("File Exists: " + file.exists());
        System.out.println("Readable: " + file.canRead());
        System.out.println("Writable: " + file.canWrite());

        if (file.exists()) {
            if (file.isFile()) {
                System.out.println("Type: Regular File");
            } else if (file.isDirectory()) {
                System.out.println("Type: Directory");
            } else {
                System.out.println("Type: Unknown");
            }
        }

        sc.close();
    }
}
```

Sample Output:

```
Enter file name with path: test.txt
File Exists: true
Readable: true
Writable: true
Type: Regular File
```

Q.2) Write a package `game` which will have 2 classes `Indoor` & `Outdoor`. Use a function `display()` to generate the list of players for the specific game. Use default & parameterized constructor. [20 Marks]

File 1: Indoor.java (inside `game` package)

```
package game;
```

```
public class Indoor {
    String gameName;
    String[] players;

    Indoor() {
        this.gameName = "Chess";
        this.players = new String[] {"Shivraj", "Ajay"};
    }

    Indoor(String gameName, String[] players) {
        this.gameName = gameName;
        this.players = players;
    }

    public void display() {
        System.out.print("Indoor Game: " + gameName + ", Players: ");
        for (String p : players) {
            System.out.print(p + " ");
        }
        System.out.println();
    }
}
```

File 2: Outdoor.java (inside `game` package)

```
package game;

public class Outdoor {
    String gameName;
    String[] players;

    Outdoor() {
        this.gameName = "Football";
        this.players = new String[] {"John", "Mike", "Sara"};
    }

    Outdoor(String gameName, String[] players) {
        this.gameName = gameName;
        this.players = players;
    }

    public void display() {
        System.out.print("Outdoor Game: " + gameName + ", Players: ");
        for (String p : players) {
            System.out.print(p + " ");
        }
        System.out.println();
    }
}
```

File 3: TestGame.java (outside package)

```
import game.Indoor;
import game.Outdoor;
```

```
public class TestGame {
    public static void main(String[] args) {
        Indoor i1 = new Indoor();
        Indoor i2 = new Indoor("Table Tennis", new String[] {"Anna", "Tom"});

        Outdoor o1 = new Outdoor();
        Outdoor o2 = new Outdoor("Cricket", new String[] {"Rahul", "Virat",
"Sachin"});

        i1.display();
        i2.display();
        o1.display();
        o2.display();
    }
}
```

Sample Output:

```
Indoor Game: Chess, Players: Shivraj Ajay
Indoor Game: Table Tennis, Players: Anna Tom
Outdoor Game: Football, Players: John Mike Sara
Outdoor Game: Cricket, Players: Rahul Virat Sachin
```

Assignment 15

Q.1) Write a Java program to print the factors of a given number. (Use Scanner class). [10 Marks]

```
import java.util.Scanner;

public class Factors {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();

        System.out.print("Factors of " + num + " are: ");
        for (int i = 1; i <= num; i++) {
            if (num % i == 0) {
                System.out.print(i + " ");
            }
        }
        sc.close();
    }
}
```

Sample Output:

```
Enter a number: 12
Factors of 12 are: 1 2 3 4 6 12
```

Q.2) Define an interface "Operation" which has methods `area()` and `volume()`. Define a constant `PI` having a value 3.142. Create a class `Circle` (member – radius), `Cylinder` (members – radius, height) which implements this interface. Calculate and display the area and volume. [20 Marks]

```
interface Operation {
    double PI = 3.142;
    double area();
    double volume();
}

class Circle implements Operation {
    double radius;

    Circle(double radius) {
        this.radius = radius;
    }

    public double area() {
        return PI * radius * radius;
    }
    public double volume() {
        return 0;
    }
}

class Cylinder implements Operation {
    double radius, height;

    Cylinder(double radius, double height) {
        this.radius = radius;
        this.height = height;
    }

    public double area() {
        return 2 * PI * radius * (radius + height);
    }

    public double volume() {
        return PI * radius * radius * height;
    }
}

public class TestOperation {
    public static void main(String[] args) {
        Circle c = new Circle(5);
        Cylinder cy = new Cylinder(5, 10);

        System.out.println("Circle Area: " + c.area());
        System.out.println("Circle Volume: " + c.volume());
        System.out.println("Cylinder Area: " + cy.area());
        System.out.println("Cylinder Volume: " + cy.volume());
    }
}
```

Sample Output:

```
Circle Area: 78.55
Circle Volume: 0.0
Cylinder Area: 471.3
Cylinder Volume: 785.5
```

Assignment 16

Q.1) Create a Hash table containing Employee name and Salary. Display the details of the hash table. [10 Marks]

```
import java.util.Hashtable;
import java.util.Enumeration;

public class EmployeeHashTable {
    public static void main(String[] args) {
        Hashtable<String, Double> employees = new Hashtable<>();

        employees.put("Shivraj", 50000.0);
        employees.put("Ajay", 60000.0);
        employees.put("Swapnil", 55000.0);
        employees.put("David", 70000.0);

        System.out.println("Employee Details:");
        Enumeration<String> names = employees.keys();
        while (names.hasMoreElements()) {
            String name = names.nextElement();
            System.out.println("Name: " + name + ", Salary: " +
employees.get(name));
        }
    }
}
```

Sample Output:

```
Employee Details:
Name: David, Salary: 70000.0
Name: Ajay, Salary: 60000.0
Name: Swapnil, Salary: 55000.0
Name: Shivraj, Salary: 50000.0
```

Q.2) Write a class Student with attributes roll no, name, age, and course. Initialize values through parameterized constructor.

- If age of student is not between 15 and 21, generate user-defined exception — "Age Not Within The Range".
- If name contains numbers or special symbols, raise exception — "Name not valid". [20 Marks]

```
class AgeNotWithinRangeException extends Exception {
    AgeNotWithinRangeException(String msg) {
        super(msg);
    }
}

class NameNotValidException extends Exception {
    NameNotValidException(String msg) {
        super(msg);
    }
}

class Student {
    int rollNo, age;
    String name, course;

    Student(int rollNo, String name, int age, String course) throws
AgeNotWithinRangeException, NameNotValidException {
        if (age < 15 || age > 21) {
            throw new AgeNotWithinRangeException("Age Not Within The Range");
        }
        if (!name.matches("[a-zA-Z ]+")) {
            throw new NameNotValidException("Name not valid");
        }
        this.rollNo = rollNo;
        this.name = name;
        this.age = age;
        this.course = course;
    }

    void display() {
        System.out.println("Roll No: " + rollNo + ", Name: " + name + ", Age: " +
age + ", Course: " + course);
    }

    public static void main(String[] args) {
        try {
            Student s1 = new Student(101, "Shivraj", 18, "BCA");
            s1.display();

            Student s2 = new Student(102, "Ajay123", 20, "BSc");
            s2.display();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }

        try {
            Student s3 = new Student(103, "Swapnil", 25, "BCom");
            s3.display();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Sample Output:

```
Roll No: 101, Name: Shivraj, Age: 18, Course: BCA
Name not valid
Age Not Within The Range
```

Assignment 17

Q.1) Write a program to accept ‘n’ integers from the user & store them in an ArrayList collection. Display the elements of ArrayList. [10 Marks]

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

public class ArrayListExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ArrayList<Integer> list = new ArrayList<>();

        System.out.print("Enter number of integers: ");
        int n = sc.nextInt();

        System.out.println("Enter " + n + " integers:");
        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();
            list.add(num);
        }

        System.out.println("Elements of ArrayList: " + list);
        sc.close();
    }
}
```

Sample Output:

```
Enter number of integers: 5
Enter 5 integers:
10
20
30
40
50
Elements of ArrayList: [10, 20, 30, 40, 50]
```

Q.2) Define a class `Patient` (patient_name, patient_age, patient_oxy_level, patient_HRCT_report).

- Handle appropriate exception while patient oxygen level < 95% **and** HRCT scan report > 10, then throw user-defined Exception “Patient is Covid Positive(+) and Need to Hospitalized”.
- Otherwise display its information. [20 Marks]

```
class CovidException extends Exception {
    CovidException(String msg) {
        super(msg);
    }
}
```

```
class Patient {
    String patient_name;
    int patient_age;
    int patient_oxy_level;
    int patient_HRCT_report;

    Patient(String patient_name, int patient_age, int patient_oxy_level, int
patient_HRCT_report) {
        this.patient_name = patient_name;
        this.patient_age = patient_age;
        this.patient_oxy_level = patient_oxy_level;
        this.patient_HRCT_report = patient_HRCT_report;
    }

    void checkHealth() throws CovidException {
        if (patient_oxy_level < 95 && patient_HRCT_report > 10) {
            throw new CovidException("Patient is Covid Positive(+) and Need to
Hospitalized");
        } else {
            System.out.println("Patient Name: " + patient_name);
            System.out.println("Age: " + patient_age);
            System.out.println("Oxygen Level: " + patient_oxy_level + "%");
            System.out.println("HRCT Report: " + patient_HRCT_report);
        }
    }

    public static void main(String[] args) {
        Patient p1 = new Patient("Shivraj", 30, 98, 5);
        Patient p2 = new Patient("Ajay", 45, 92, 12);

        try {
            p1.checkHealth();
        } catch (CovidException e) {
            System.out.println(e.getMessage());
        }

        try {
            p2.checkHealth();
        } catch (CovidException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Sample Output:

```
Patient Name: Shivraj
Age: 30
Oxygen Level: 98%
HRCT Report: 5
Patient is Covid Positive(+) and Need to Hospitalized
```

Assignment 18

Q.1) Write a Java program to accept a number from user, if it is zero then throw user-defined Exception “Number Is Zero”, otherwise calculate the sum of first and last digit of that number. (Use static keyword) [10 Marks]

```
import java.util.Scanner;

class ZeroNumberException extends Exception {
    ZeroNumberException(String msg) {
        super(msg);
    }
}

public class SumFirstLastDigit {
    static int sumFirstLast(int num) {
        int last = num % 10;
        int first = num;
        while (first >= 10) {
            first /= 10;
        }
        return first + last;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();

        try {
            if (num == 0) {
                throw new ZeroNumberException("Number Is Zero");
            }
            int sum = sumFirstLast(num);
            System.out.println("Sum of first and last digit: " + sum);
        } catch (ZeroNumberException e) {
            System.out.println(e.getMessage());
        }

        sc.close();
    }
}
```

Sample Output 1:

```
Enter a number: 1234
Sum of first and last digit: 5
```

Sample Output 2:

```
Enter a number: 0
Number Is Zero
```

Q.2) Write a program to read the contents of “abc.txt” file. Display the contents of file in uppercase as output. [10 Marks]

```
import java.io.*;

public class ReadFileUpperCase {
    public static void main(String[] args) {
        try {
            File file = new File("abc.txt");
            if (!file.exists()) {
                System.out.println("File does not exist.");
                return;
            }

            BufferedReader br = new BufferedReader(new FileReader(file));
            String line;
            System.out.println("Contents of file in uppercase:");
            while ((line = br.readLine()) != null) {
                System.out.println(line.toUpperCase());
            }
            br.close();
        } catch (IOException e) {
            System.out.println("Error reading file: " + e.getMessage());
        }
    }
}
```

Sample Output (if abc.txt contains “Hello World”):

```
Contents of file in uppercase:
HELLO WORLD
```

Assignment 19

Q.1) Write a Java program to print the factors of a given number. (Use Scanner class). [10 Marks]

```
import java.util.Scanner;

public class Factors {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();

        System.out.print("Factors of " + num + " are: ");
        for (int i = 1; i <= num; i++) {
            if (num % i == 0) {
                System.out.print(i + " ");
            }
        }

        sc.close();
    }
}
```

Sample Output:

```
Enter a number: 12
Factors of 12 are: 1 2 3 4 6 12
```

Q.2) Write a Java program to accept Doctor Name from the user and check whether it is valid or not.

- It should not contain digits or special symbols.
- If invalid, throw user-defined Exception – “Name is Invalid”, otherwise display it. [20 Marks]

```
import java.util.Scanner;

class NameInvalidException extends Exception {
    NameInvalidException(String msg) {
        super(msg);
    }
}

public class DoctorNameCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Doctor Name: ");
        String name = sc.nextLine();

        try {
            if (!name.matches("[a-zA-Z ]+")) {
                throw new NameInvalidException("Name is Invalid");
            }
            System.out.println("Doctor Name: " + name);
        } catch (NameInvalidException e) {
            System.out.println(e.getMessage());
        }

        sc.close();
    }
}
```

Sample Output 1 (Valid Name):

```
Enter Doctor Name: John Smith
Doctor Name: John Smith
```

Sample Output 2 (Invalid Name):

```
Enter Doctor Name: Dr123
Name is Invalid
```

Assignment 20

Q.1) Write a class `Driver` with attributes `license_no`, `name`, `address` and `age`. Initialize values through the parameterized constructor. If age of Driver is less than 18 then user-defined exception should be generated —“Age is below 18 years—”. [10 Marks]

```
import java.util.Scanner;

class AgeBelow18Exception extends Exception {
    AgeBelow18Exception(String msg) {
        super(msg);
    }
}

class Driver {
    String license_no, name, address;
    int age;

    Driver(String license_no, String name, String address, int age) throws
AgeBelow18Exception {
        if (age < 18) {
            throw new AgeBelow18Exception("Age is below 18 years");
        }
        this.license_no = license_no;
        this.name = name;
        this.address = address;
        this.age = age;
    }

    void display() {
        System.out.println("License No: " + license_no);
        System.out.println("Name: " + name);
        System.out.println("Address: " + address);
        System.out.println("Age: " + age);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter License No: ");
        String license_no = sc.nextLine();
        System.out.print("Enter Name: ");
        String name = sc.nextLine();
        System.out.print("Enter Address: ");
        String address = sc.nextLine();
        System.out.print("Enter Age: ");
        int age = sc.nextInt();

        try {
            Driver d = new Driver(license_no, name, address, age);
            d.display();
        } catch (AgeBelow18Exception e) {
            System.out.println(e.getMessage());
        }
        sc.close();
    }
}
```

Sample Output 1 (Valid Age):

```
Enter License No: DL1234
Enter Name: John
Enter Address: Pune
Enter Age: 25
License No: DL1234
Name: John
Address: Pune
Age: 25
```

Sample Output 2 (Invalid Age):

```
Enter License No: DL5678
Enter Name: Shivraj
Enter Address: Mumbai
Enter Age: 16
Age is below 18 years
```

Q.2) Construct a linked list containing names of colours: red, blue, yellow, and orange. Then extend the program to do the following:

i. Display the contents of the list using an `Iterator`
ii. Display the contents of the list in reverse order using a `ListIterator`
iii. Create another list containing pink and green. Insert the elements of this list between blue and yellow
[20 Marks]

```
import java.util.*;

public class ColorList {
    public static void main(String[] args) {
        LinkedList<String> colors = new LinkedList<>();
        colors.add("red");
        colors.add("blue");
        colors.add("yellow");
        colors.add("orange");

        // i. Display using Iterator
        System.out.println("Colors using Iterator:");
        Iterator<String> it = colors.iterator();
        while (it.hasNext()) {
            System.out.print(it.next() + " ");
        }
        System.out.println();

        // ii. Display in reverse using ListIterator
        System.out.println("Colors in reverse using ListIterator:");
        ListIterator<String> lit = colors.listIterator(colors.size());
        while (lit.hasPrevious()) {
            System.out.print(lit.previous() + " ");
        }
        System.out.println();

        // iii. Insert another list between blue and yellow
        LinkedList<String> newColors = new LinkedList<>();
        newColors.add("pink");
        newColors.add("green");

        int index = colors.indexOf("yellow");
        colors.addAll(index, newColors);

        System.out.println("Colors after inserting new list:");
        for (String c : colors) {
            System.out.print(c + " ");
        }
    }
}
```

Sample Output:

```
Colors using Iterator:
red blue yellow orange
Colors in reverse using ListIterator:
orange yellow blue red
Colors after inserting new list:
red blue pink green yellow orange
```