

JAVA PROGRAMMING LAB

BCA-DS-452A

**Manav Rachna International Institute of Research and
Studies**

Department of Computer Applications

Submitted By	
Student Name	Purujit malik
Roll No	23/SCA/BCA(AI-ML)/35
Programme	BCA(AI-ML)
Semester	4th Semester
Section/Group	4-C
Department	School of Computer Applications
Batch	2023-26
Submitted To	
Faculty Name	Mrs.priyanka sharma

S.No	Program name	date
	LAB SESSION 1	
1.	Write java program to print hello world	
2.	Java Program to take input from user and print the sum of two numbers	
3.	Create a java program to check whether a number entered by user is even or odd	
4.	Create a java program to print the average and sum of 5 numbers entered by user.	
5.	Program to calculate the factorial of a number	
6.	Program to print Fibonacci series up to n terms	
7.	LAB SESSION 2	
1.	Program to reverse a number	
2.	Write a program to find the average and sum of the N numbers using Command line argument.	
3.	Program to check if a number is a palindrome	
4.	Program for a simple calculator	
5.	Program to check if a number is prime	
6.	Program to check if a number is an Armstrong number	
	LAB SESSION 3	
	Print Multiplication Table	
	Calculate Sum and Average of Array Elements	
	Reverse a String	
2	Find Factorial of a Number Using Recursion	
3	Sort an Array in Ascending Order	
	LAB SESSION 4	
4	Check Palindrome for a String	
5	Count Vowels and Consonants in a String	
6	Implement a Simple Banking System	
7	Write a program to demonstrate type casting.	
8	Write a program to generate prime numbers between 1 & given number	
9	LAB SESSION 5	
10	Program to Demonstrate a Simple Class with Methods	
11	Program for Class with Parameterized Constructor	
12	Program to Find the Area of a Rectangle Using Methods	
13	Program for Bank Account Class with Deposit and Withdraw Methods	

14	Program to Demonstrate Method Overloading	
15	LAB SESSION 5	
16	Program to Demonstrate Static Methods	
17	Program to Demonstrate Method Overriding	
18	Program to Demonstrate Getters and Setters	
19	Program to Demonstrate a Class with Multiple Methods	
20	Program to Demonstrate Object Passing in Methods	
21	Write a program to create a simple class to find out the area and perimeter of rectangle using super and this keyword.	
22	Write a program to count the number of objects created for a class using static member function	

Lab Session 1 file

1) Write a java program to print hello world

Input:

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

Output:

```
Hello World!
```

```
=== Code Execution Successful ===
```

2. Java Program to take input from the user and print the sum of two numbers.

Input:

```
import java.util.Scanner;  
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner=new Scanner(System.in);  
        System.out.println("Enter two numbers");  
        int a=scanner.nextInt();  
        int b=scanner.nextInt();  
        int sum=a+b;  
        System.out.println("Sum is: " + sum);  
    }  
}
```

Output:

```
Enter two numbers  
15  
32  
Sum is: 47
```

Create a java program to check whether a number entered by the user is even or odd.

Input:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter a number");
        int a=scanner.nextInt();
        if(a%2==0)
            System.out.println(a+" is even");
        else
            System.out.println(a+" is odd");
    }
}
```

Output:

```
Enter a number
18
18 is even
```

3. Create a java program to print the average and sum of 5 numbers entered by the user.

Input:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter 5 numbers");
        int a=scanner.nextInt();
        int b=scanner.nextInt();
        int c=scanner.nextInt();
        int d=scanner.nextInt();
        int e=scanner.nextInt();
        int sum=a+b+c+d+e;
        int average=sum/5;
        System.out.println("Sum is: "+sum);
        System.out.println("Average is: "+average);
    }
}
```

Output:

```
Enter 5 numbers
12
32
65
45
78
Sum is: 232
Average is: 46
```

4. Program to calculate the factorial of a number.

Input:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        int factorial=1;
        System.out.println("Enter a number");
        int a=scanner.nextInt();
        for(int i=a;i>=1;i--)
        {
            factorial=factorial*i;
        }
        System.out.println("Factorial is: "+factorial);
    }
}
```

Output:

```
Enter a number
15
Factorial is: 2004310016
```

5. Program to print Fibonacci series up to n terms.

Input:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        int a=0,b=1,c;
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter a number");
        int n=scanner.nextInt();
        System.out.print(a);
        System.out.print(b);
        for(int i=3;i<=n;i++)
        {
            c=a+b;
            System.out.print(c);
            a=b;
            b=c;
        }
    }
}
```

Output:

```
Enter a number
4
0112
```


Lab session 2

1. Program to reverse a number

Input:

```
import java.util.Scanner;

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

        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        int reversed = 0;
        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num = num / 10;
        }
        System.out.println("Reversed number: " + reversed);
        scanner.close();
    }
}
```

Output:

```
Enter a number: 15
Reversed number: 51
```

2. Java Program to check if a number is a palindrome

Input:

```
import java.util.Scanner;

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

        System.out.print("Enter a number: ");
        int num = scanner.nextInt();

        int originalNum = num;
        int reversed = 0;

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

        if (originalNum == reversed) {
            System.out.println(originalNum + " is a Palindrome.");
        } else {
            System.out.println(originalNum + " is NOT a Palindrome.");
        }
        scanner.close();
    }
}
```

Output:

```
Enter a number: 14
14 is NOT a Palindrome.
```

Create a java Program for a simple calculator

Input:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter first number: ");
        double num1 = scanner.nextDouble();
        System.out.print("Enter an operator (+, -, *, /): ");
        char operator = scanner.next().charAt(0);
        System.out.print("Enter second number: ");
        double num2 = scanner.nextDouble();
        double result;
        switch (operator) {
            case '+':
                result = num1 + num2;
                System.out.println("Result: " + result);
                break;
            case '-':
                result = num1 - num2;
                System.out.println("Result: " + result);
                break;
            case '*':
                result = num1 * num2;
                System.out.println("Result: " + result);
                break;
            case '/':
                if (num2 != 0) {
                    result = num1 / num2;
                    System.out.println("Result: " + result);
                } else {
                    System.out.println("Error! Division by zero is not allowed.");
                }
                break;
            default:
                System.out.println("Invalid operator!");
        }
        scanner.close();
    }
}
```

Output:

```
Enter first number: 15
Enter an operator (+, -, *, /): -
Enter second number: 85
Result: -70.0
```

3. Create a java Program to check if a number is prime

Input:

```
import java.util.Scanner;

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

Output:

```
Enter a number: 18
18 is NOT a Prime Number.
```

4. Program to check if a number is an Armstrong number

Input:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        int originalNum = num;
        int sum = 0;
        int digits = String.valueOf(num).length();
        while (num != 0) {
            int digit = num % 10;
            sum += Math.pow(digit, digits);
            num /= 10;
        }
        if (sum == originalNum) {
            System.out.println(originalNum + " is an Armstrong Number.");
        } else {
            System.out.println(originalNum + " is NOT an Armstrong Number.");
        }
        scanner.close();
    }
}
```

Output:

```
Enter a number: 56
56 is NOT an Armstrong Number.
```

5. Program to Find the Largest of Three Numbers using ternary operator

Input:

```
import java.util.Scanner;

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

        System.out.print("Enter first number: ");
        int num1 = scanner.nextInt();

        System.out.print("Enter second number: ");
        int num2 = scanner.nextInt();

        System.out.print("Enter third number: ");
        int num3 = scanner.nextInt();

        int largest = (num1 > num2) ?
            ((num1 > num3) ? num1 : num3) :
            ((num2 > num3) ? num2 : num3);

        System.out.println("The largest number is: " + largest);

        scanner.close();
    }
}
```

Output:

```
Enter first number: 25
Enter second number: 52
Enter third number: 32
The largest number is: 52
```

```

1 import java.util.Scanner;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter a string: ");
7
8         String input = scanner.nextLine();
9         countVowelsAndConsonants(input);
10        scanner.close();
11    }
12    public static void countVowelsAndConsonants(String str) {
13        int vowelCount = 0, consonantCount = 0;
14        str = str.toLowerCase();
15        for (int i = 0; i < str.length(); i++) {
16            char ch = str.charAt(i);
17            if (Character.isLetter(ch)) {
18                if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
19                    vowelCount++;
20                }
21                else {
22                    consonantCount++;
23                }
24            }
25        }
26        System.out.println("Vowels: " + vowelCount);
27        System.out.println("Consonants: " + consonantCount);
28    }
29 }
30
31

```

Output:

STDIN

qwerty

Output:

Enter a string: Vowels: 1

Consonants: 5

```

import java.util.Scanner;

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

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        if (isPalindrome(input)) {
            System.out.println(input + " is a palindrome.");
        } else {
            System.out.println(input + " is not a palindrome.");
        }

        scanner.close();
    }

    public static boolean isPalindrome(String str) {
        str = str.toLowerCase();

        int left = 0, right = str.length() - 1;

        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }

        return true;
    }
}

```

output:

STDIN

123321

Output:

Enter a string: 123321 is a palindrome.

1.write a program to demonstrate type casting in java

```
1 public class TypeCastingDemo {
2     public static void main(String[] args) {
3
4         int intValue = 100;
5         double doubleValue = intValue;
6         System.out.println("Implicit Casting:");
7         System.out.println("Integer Value: " + intValue);
8         System.out.println("Double Value after implicit casting: " + doubleValue);
9
10
11         double anotherDoubleValue = 9.78;
12         int anotherIntValue = (int) anotherDoubleValue;
13         System.out.println("\nExplicit Casting:");
14         System.out.println("Double Value: " + anotherDoubleValue);
15         System.out.println("Integer Value after explicit casting: " + anotherIntValue);
16
17
18         Animal animal = new Dog();
19         animal.makeSound();
20
21
22         Dog dog = (Dog) animal;
23         dog.fetch();
24     }
25 }
26
27
28 class Animal {
29     void makeSound() {
30         System.out.println("Animal makes a sound");
31     }
32 }
33
34
35 class Dog extends Animal {
36     void makeSound() {
37         System.out.println("Dog barks");
38     }
39
40     void fetch() {
41         System.out.println("Dog fetches the ball");
42     }
43 }
```

Output:

Output:

Implicit Casting:

Integer Value: 100

Double Value after implicit casting: 100.0

Explicit Casting:

Double Value: 9.78

Integer Value after explicit casting: 9

Dog barks

Dog fetches the ball

2.write a program to generate prime numbers between 1 & given number

```
1 import java.util.Scanner;
2
3 public class PrimeNumberGenerator {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7
8         System.out.print("Enter a number: ");
9         int number = scanner.nextInt();
10
11         System.out.println("Prime numbers between 1 and " + number + " are:");
12
13
14         for (int i = 2; i <= number; i++) {
15             if (isPrime(i)) {
16                 System.out.print(i + " ");
17             }
18         }
19
20
21         scanner.close();
22     }
23
24
25     public static boolean isPrime(int num) {
26         if (num <= 1) {
27             return false;
28         }
29         for (int i = 2; i <= Math.sqrt(num); i++) {
30             if (num % i == 0) {
31                 return false;
32             }
33         }
34         return true;
35     }
36 }
```

Output:

STDIN

20

Output:

Enter a number: Prime numbers between 1 and 20 are:
2 3 5 7 11 13 17 19

1. program to demonstrate a simple class with methods in java

```
1
2 ▾ class Car {
3
4     String make;
5     String model;
6     int year;
7
8
9 ▾     public Car(String make, String model, int year) {
10         this.make = make;
11         this.model = model;
12         this.year = year;
13     }
14
15
16 ▾     public void displayDetails() {
17         System.out.println("Car Make: " + make);
18         System.out.println("Car Model: " + model);
19         System.out.println("Car Year: " + year);
20     }
21
22
23 ▾     public void start() {
24         System.out.println("The car is starting...");
25     }
26
27
28 ▾     public void stop() {
29         System.out.println("The car is stopping...");
```

```
27
28 ▾ public void stop() {
29     System.out.println("The car is stopping...");
30 }
31 }
32
33
34 ▾ public class Main {
35     public static void main(String[] args) {
36
37         Car myCar = new Car("Toyota", "Corolla", 2020);
38
39
40         myCar.displayDetails();
41         myCar.start();
42         myCar.stop();
43     }
44 }
45
```

Output

```
Car Make: Toyota
Car Model: Corolla
Car Year: 2020
The car is starting...
The car is stopping...
```

```
=== Code Execution Successful ===
```

2. program to find the area of rectangle using methods




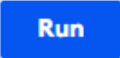
```
1
2 class Rectangle {
3
4     double length;
5     double width;
6
7     public Rectangle(double length, double width) {
8         this.length = length;
9         this.width = width;
10    }
11    public double calculateArea() {
12        return length * width;
13    }
14
15    public void displayArea() {
16        System.out.println("The area of the rectangle is: " + calculateArea() + "
17                               square units.");
18    }
19 }
20 public class Main {
21     public static void main(String[] args) {
22         Rectangle myRectangle = new Rectangle(5.0, 3.0);
23
24
25         myRectangle.displayArea();
26     }
27 }
28
```

Output

The area of the rectangle is: 15.0 square units.

=== Code Execution Successful ===

1. program to demonstrate Method Overloading in java

```
Main.java    Share  Run

1 class Calculator {
2     public int add(int a, int b) {
3         return a + b;
4     }
5     public int add(int a, int b, int c) {
6         return a + b + c;
7     }
8     public double add(double a, double b) {
9         return a + b;
10    }
11 }
12 public class Main {
13     public static void main(String[] args) {
14
15         Calculator calc = new Calculator();
16
17         System.out.println("Sum of 2 integers: " + calc.add(5, 10));
18
19         System.out.println("Sum of 3 integers: " + calc.add(5, 10, 15));
20
21         System.out.println("Sum of 2 doubles: " + calc.add(5.5, 10.5));
22     }
23 }
24
```

Output

```
Sum of 2 integers: 15
Sum of 3 integers: 30
Sum of 2 doubles: 16.0
```

```
=== Code Execution Successful ===
```

2. Program for class with parameterized constructor

```
1 ▾ class Student {  
2     String name;  
3     int age;  
4  
5 ▾     public Student(String name, int age) {  
6         this.name = name;  
7         this.age = age;  
8     }  
9  
10 ▾    public void displayDetails() {  
11        System.out.println("Student Name: " + name);  
12        System.out.println("Student Age: " + age);  
13    }  
14 }  
15  
16 ▾ public class Main {  
17 ▾     public static void main(String[] args) {  
18  
19         Student student1 = new Student("Alice", 20);  
20         Student student2 = new Student("Bob", 22);  
21  
22         student1.displayDetails();  
23         student2.displayDetails();  
24     }  
25 }  
26
```

Output

```
Student Name: Alice  
Student Age: 20  
Student Name: Bob  
Student Age: 22
```

```
=== Code Execution Successful ===|
```


3. Program for Bank Account Class with Deposit and Withdraw Methods in java

```
1 - class BankAccount {
2     private double balance;
3 - public BankAccount(double initialBalance) {
4 -     if (initialBalance >= 0) {
5         balance = initialBalance;
6 -     } else {
7         balance = 0;
8         System.out.println("Initial balance can't be negative.
           Setting balance to 0.");
9     }
10 }
11 - public void deposit(double amount) {
12 -     if (amount > 0) {
13         balance += amount;
14         System.out.println("Deposited: $" + amount);
15 -     } else {
16         System.out.println("Deposit amount must be positive.");
17     }
18 }
19
20 - public void withdraw(double amount) {
21 -     if (amount > 0) {
22 -         if (balance >= amount) {
23             balance -= amount;
24             System.out.println("Withdrew: $" + amount);
25 -         } else {
```

```
25 ▾         } else {
26             System.out.println("Insufficient funds. Unable to
                withdraw $" + amount);
27         }
28 ▾     } else {
29         System.out.println("Withdrawal amount must be positive.");
30     }
31 }
32
33 ▾ public void checkBalance() {
34     System.out.println("Current Balance: $" + balance);
35 }
36 }
37
38 ▾ public class Main {
39 ▾     public static void main(String[] args) {
40         BankAccount myAccount = new BankAccount(500.0);
41
42         myAccount.checkBalance();
43
44         myAccount.deposit(200.0);
45         myAccount.checkBalance();
46
47         myAccount.withdraw(100.0);
48         myAccount.checkBalance();
49
50         myAccount.withdraw(700.0);
51         myAccount.checkBalance();
52     }
53 }
54
```

Output

```
Current Balance: $500.0  
Deposited: $200.0  
Current Balance: $700.0  
Withdrew: $100.0  
Current Balance: $600.0  
Insufficient funds. Unable to withdraw $700.0  
Current Balance: $600.0
```

```
=== Code Execution Successful ===|
```

1. Program to Demonstrate Static Methods in java

```
1 public class Main {  
2  
3     static int counter = 0;  
4  
5     public static void incrementCounter() {  
6         counter++;  
7         System.out.println("Counter: " + counter);  
8     }  
9  
10    public void displayMessage() {  
11        System.out.println("This is a non-static method.");  
12    }  
13  
14    public static void main(String[] args) {  
15        Main.incrementCounter();  
16        Main.incrementCounter();  
17  
18        Main obj = new Main();  
19        obj.displayMessage();  
20    }  
21 }
```

Output

```
Counter: 1  
Counter: 2  
This is a non-static method.
```

```
=== Code Execution Successful ===
```

2. Program to demonstrate method overriding in java

```

1 class Animal {
2     public void sound() {
3         System.out.println("Animal makes a sound");
4     }
5 }
6 class Dog extends Animal {
7     @Override
8     public void sound() {
9         System.out.println("Dog barks");
10    }
11 }
12
13 class Cat extends Animal {
14     @Override
15     public void sound() {
16         System.out.println("Cat meows");
17     }
18 }
19 public class Main {
20     public static void main(String[] args) {
21         Animal myDog = new Dog();
22         Animal myCat = new Cat();
23         myDog.sound();
24         myCat.sound();
25     }
26 }

```

Output

```

Dog barks
Cat meows

```

```

=== Code Execution Successful ===

```

3. Write a program to create a simple class to find out the area and perimeter of rectangle using super keyword

```
1 class Shape {
2     double length, width;
3
4     public Shape(double length, double width) {
5         this.length = length;
6         this.width = width;
7     }
8     public double calculateArea() {
9         return length * width;
10    }
11    public double calculatePerimeter() {
12        return 2 * (length + width);
13    }
14 }
15
16 class Rectangle extends Shape {
17
18     public Rectangle(double length, double width) {
19         super(length, width);
20     }
21     public void display() {
22         System.out.println("Area of Rectangle: " + calculateArea());
23         System.out.println("Perimeter of Rectangle: " +
24             calculatePerimeter());
25     }
26 }
```

```
26 public class Main {
27     public static void main(String[] args) {
28         Rectangle rectangle = new Rectangle(5, 3);
29         rectangle.display();
30     }
31 }
```

Output

```
Area of Rectangle: 15.0
Perimeter of Rectangle: 16.0
```

```
=== Code Execution Successful ===
```

LAB SESSION 6

LAB SESSION 6
Program to Demonstrate Static Methods
Program to Demonstrate Method Overriding
Program to Demonstrate Getters and Setters
Program to Demonstrate a Class with Multiple Methods
Program to Demonstrate Object Passing in Methods
Write a program to create a simple class to find out the area and perimeter of rectangle using super and this keyword.

Solutions :

1.

```
1- public class Main {
2-     public static void main(String[] args) {
3-         // Calling static methods without creating an object of the MathOperations class
4-         System.out.println("Sum of two integers: " + MathOperations.add(5, 10));
5-         System.out.println("Product of two integers: " + MathOperations.multiply(5, 10));
6-     }
7- }
8
9- class MathOperations {
10-     // Static method to add two integers
11-     public static int add(int a, int b) {
12-         return a + b;
13-     }
14
15-     // Static method to multiply two integers
16-     public static int multiply(int a, int b) {
17-         return a * b;
18-     }
19- }
```

```
Sum of two integers: 15
Product of two integers: 50
```

2.

```
1 public class Main {
2     public static void main(String[] args) {
3         // Creating an object of the Dog class
4         Animal myDog = new Dog();
5
6         // Creating an object of the Cat class
7         Animal myCat = new Cat();
8
9         // Calling the overridden methods
10        myDog.sound();
11        myCat.sound();
12    }
13 }
14
15 class Animal {
16     // Method to be overridden
17     public void sound() {
18         System.out.println("Animal makes a sound");
19     }
20 }
21
22 class Dog extends Animal {
23     // Overriding the sound method
24     @Override
25     public void sound() {
26         System.out.println("Dog barks");
27     }
28 }
29
30 class Cat extends Animal {
31     // Overriding the sound method
32     @Override
33     public void sound() {
34         System.out.println("Cat meows");
35     }
36 }
```

Dog barks

Cat meows

3.

```
1 public class Main {
2     public static void main(String[] args) {
3         // Creating an object of the Person class
4         Person person = new Person();
5
6         // Setting values using setters
7         person.setName("Alice");
8         person.setAge(30);
9
10        // Getting values using getters
11        System.out.println("Name: " + person.getName());
12        System.out.println("Age: " + person.getAge());
13    }
14 }
15
16 class Person {
17     // Private fields
18     private String name;
19     private int age;
20
21     // Getter for name
22     public String getName() {
23         return name;
24     }
25
26     // Setter for name
27     public void setName(String name) {
28         this.name = name;
29     }
30
31     // Getter for age
32     public int getAge() {
33         return age;
34     }
35
36     // Setter for age
37     public void setAge(int age) {
38         this.age = age;
39     }
40 }
```

```
Name: Alice
Age: 30
```

4.

```
1 public class Main {
2     public static void main(String[] args) {
3         // Creating an object of the Calculator class
4         Calculator calc = new Calculator();
5
6         // Calling various methods on the Calculator object
7         System.out.println("Addition: " + calc.add(10, 5));
8         System.out.println("Subtraction: " + calc.subtract(10, 5));
```

```
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
```

5.

```
1+ public class Main {
2+     public static void main(String[] args) {
3+         // Creating an object of the Rectangle class
4+         Rectangle rect1 = new Rectangle(10, 5);
5+
6+         // Displaying the area of the rectangle using a method
7+         displayArea(rect1);
8+
9+         // Modifying the rectangle dimensions using a method
10+        modifyRectangle(rect1, 15, 7);
11+
12+        // Displaying the modified area of the rectangle
```

```

39- public double getLength() {
40-     return length;
41- }
42-
43- // Setter for length
44- public void setLength(double length) {
45-     this.length = length;
46- }
47-
48- // Getter for width
49- public double getWidth() {
50-     return width;
51- }
52-
53- // Setter for width
54- public void setWidth(double width) {
55-     this.width = width;
56- }
57-
58- // Method to calculate the area of the rectangle
59- public double calculateArea() {
60-     return length * width;
61- }
62- }

```

Area of the rectangle: 50.0
Area of the rectangle: 105.0

6.

```

1- public class Main {
2-     public static void main(String[] args) {
3-         // Creating an object of the Rectangle class
4-         Rectangle rect = new Rectangle(10, 5);
5-
6-         // Displaying the area and perimeter of the rectangle
7-         System.out.println("Area: " + rect.calculateArea());
8-         System.out.println("Perimeter: " + rect.calculatePerimeter());
9-     }
10- }
11-
12- class Shape {
13-     protected double length;

```

LAB SESSION 7

Q1) Program to Demonstrate Static Methods

Sol:

```
class StaticDemo {  
    // Static method to multiply two integers  
    static int multiply(int a, int b) {  
        return a * b;  
    }  
  
    // Non-static method to display a message  
    void display() {  
        System.out.println("Static method example.");  
    }  
  
    public static void main(String[] args) {  
        // Calling the static method without creating an instance of the class  
        int result = StaticDemo.multiply(5, 10);  
        System.out.println("Product of two integers: " + result);  
  
        // Creating an instance of the class to call the non-static method  
        StaticDemo demo = new StaticDemo();  
        demo.display();  
    }  
}
```

Output:

```
Enter the first integer: 7  
Enter the second integer: 8  
Product of two integers: 56
```

Q2) Program to Demonstrate Method Overriding

Sol:

```

class Animal {
    // Method in the superclass
    void sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    // Overriding the method in the subclass
    @Override
    void sound() {
        System.out.println("Dog barks");
    }
}

class Cat extends Animal {
    // Overriding the method in the subclass
    @Override
    void sound() {
        System.out.println("Cat meows");
    }
}

public class MethodOverridingDemo {
    public static void main(String[] args) {
        Animal myAnimal = new Animal(); // Creating an instance of Animal
        Animal myDog = new Dog();        // Creating an instance of Dog
        Animal myCat = new Cat();         // Creating an instance of Cat

        myAnimal.sound(); // Calls the sound method from Animal class
        myDog.sound();    // Calls the sound method from Dog class
        myCat.sound();    // Calls the sound method from Cat class
    }
}

```

Output:

```

Animal makes a sound
Dog barks
Cat meows

```

Q3) Program to Demonstrate Getters and Setters

Sol:

```

class Person {
    // Private fields
    private String name;
    private int age;

    // Getter for name
    public String getName() {
        return name;
    }

    // Setter for name
    public void setName(String name) {
        this.name = name;
    }

    // Getter for age
    public int getAge() {
        return age;
    }

    // Setter for age
    public void setAge(int age) {
        if (age > 0) { // Adding a basic validation check
            this.age = age;
        } else {
            System.out.println("Age must be positive");
        }
    }
}

public class GettersSettersDemo {
    public static void main(String[] args) {
        Person person = new Person();

        // Setting values using setters
        person.setName("Alice");
        person.setAge(25);

        // Getting values using getters
        System.out.println("Name: " + person.getName());
        System.out.println("Age: " + person.getAge());

        // Trying to set an invalid age
        person.setAge(-5);
        System.out.println("Age after setting invalid value: " + person.getAge());
    }
}

```

Output:

```

Name: Alice
Age: 25
Age must be positive
Age after setting invalid value: 25

```

Q4) Program to Demonstrate a Class with Multiple Methods

Sol:

```
class Calculator {
    // Method for addition
    int add(int a, int b) {
        return a + b;
    }

    // Method for subtraction
    int subtract(int a, int b) {
        return a - b;
    }

    // Method for multiplication
    int multiply(int a, int b) {
        return a * b;
    }

    // Method for division
    double divide(int a, int b) {
        if (b != 0) {
            return (double) a / b;
        } else {
            System.out.println("Division by zero is not allowed.");
            return 0;
        }
    }
}

public class MultipleMethodsDemo {
    public static void main(String[] args) {
        Calculator calculator = new Calculator();

        int a = 10;
        int b = 5;

        System.out.println("Addition: " + calculator.add(a, b));
        System.out.println("Subtraction: " + calculator.subtract(a, b));
        System.out.println("Multiplication: " + calculator.multiply(a, b));
        System.out.println("Division: " + calculator.divide(a, b));
    }
}
```

Output:

```
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
```

Q5) Program to Demonstrate Object Passing in Methods

Sol:

```
class Rectangle {
    private double length;
    private double width;

    // Setter for length
    public void setLength(double length) {
        this.length = length;
    }

    // Setter for width
    public void setWidth(double width) {
        this.width = width;
    }

    // Getter for length
    public double getLength() {
        return length;
    }

    // Getter for width
    public double getWidth() {
        return width;
    }
}

class AreaCalculator {
    // Method to calculate the area of a Rectangle object
    public double calculateArea(Rectangle rectangle) {
        return rectangle.getLength() * rectangle.getWidth();
    }
}

public class ObjectPassingDemo {
    public static void main(String[] args) {
        Rectangle myRectangle = new Rectangle();
        myRectangle.setLength(5.0);
        myRectangle.setWidth(3.0);

        AreaCalculator calculator = new AreaCalculator();
        double area = calculator.calculateArea(myRectangle);

        System.out.println("The area of the rectangle is: " + area);
    }
}
```

Output:

```
The area of the rectangle is: 15.0
```

Q6) Write a program to create a simple class to find out the area and perimeter of rectangle using super and this keyword.\

Sol:

```
class Shape {
    double length;
    double width;

    Shape(double length, double width) {
        this.length = length;
        this.width = width;
    }

    // Method to calculate the area
    double area() {
        return length * width;
    }

    // Method to calculate the perimeter
    double perimeter() {
        return 2 * (length + width);
    }
}

class Rectangle extends Shape {
    Rectangle(double length, double width) {
        // Using super keyword to call the constructor of the superclass
        super(length, width);
    }

    // Method to display area and perimeter
    void display() {
        System.out.println("Area: " + this.area());
        System.out.println("Perimeter: " + this.perimeter());
    }
}

public class SuperAndThisDemo {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(5.0, 3.0);

        // Displaying the area and perimeter using the display method
        rectangle.display();
    }
}
```

Output:

```
Area: 15.0
Perimeter: 16.0
```

Lab Session 8

1. Write a Java program demonstrating String methods like `substring()`, `replace()`, and `split()`.
Input:

```
public class StringMethodsDemo {  
    public static void main(String[] args) {  
  
        String sentence = "Java is a powerful programming language";  
  
        String part = sentence.substring(10, 18);  
        System.out.println("Substring (10, 18): " + part);  
  
        String replaced = sentence.replace("Java", "Python");  
        System.out.println("After replace: " + replaced);  
  
        String[] words = sentence.split(" ");  
        System.out.println("After split:");  
        for (String word : words) {  
            System.out.println(word);  
        }  
    }  
}
```

Output:

Output

```
Substring (10, 18): powerful  
After replace: Python is a powerful programming language  
After split:  
Java  
is  
a  
powerful  
programming  
language  
  
=== Code Execution Successful ===
```

2. Create a custom exception Age Exception that checks if a person's age is valid (above 18). in java

Input:

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

    public static void checkAge(int age) throws AgeException {
        if (age <= 18) {
            throw new AgeException("Age is not valid. Must be above 18.");
        } else {
            System.out.println("Age is valid: " + age);
        }
    }

    public static void main(String[] args) {
        int[] sampleAges = {16, 18, 21};

        for (int age : sampleAges) {
            System.out.println("Checking age: " + age);
            try {
                checkAge(age);
            } catch (AgeException e) {
                System.out.println("Exception: " + e.getMessage());
            }
            System.out.println();
        }
    }
}
```

Output:

Output

```
Checking age: 16
Exception: Age is not valid. Must be above 18.

Checking age: 18
Exception: Age is not valid. Must be above 18.

Checking age: 21
Age is valid: 21
```

3. Create a Java program that demonstrates various string functions and string handling techniques in Java. This program includes everyday operations like: Length of a string, Concatenation, Character extraction, Substring, Searching, String comparison, Changing case, Trimming, Replacing, Splitting

Input:

```

public class StringFunctionsDemo {
    public static void main(String[] args) {
        String str1 = " Hello World ";
        String str2 = "Java Programming";

        System.out.println("Length of str1: " + str1.length());

        String concatStr = str1.trim().concat(" - ").concat(str2);
        System.out.println("Concatenated String: " + concatStr);

        System.out.println("Character at index 1 in str2: " + str2.charAt(1));

        System.out.println("Substring of str2 (5 to 16): " + str2.substring(5, 16));

        System.out.println("Index of 'World' in str1: " + str1.indexOf("World"));

        System.out.println("str1 equals str2? " + str1.equals(str2));
        System.out.println("str1 equalsIgnoreCase 'hello world'? " + str1.trim()
            ().equalsIgnoreCase("hello world"));

        System.out.println("str2 in uppercase: " + str2.toUpperCase());
        System.out.println("str2 in lowercase: " + str2.toLowerCase());

        System.out.println("str1 trimmed: '" + str1.trim() + "'");

        System.out.println("str2 with 'Java' replaced by 'Python': " + str2.replace("Java",
            "Python"));

        String[] words = str2.split(" ");
        System.out.println("Words in str2:");
        for (String word : words) {
            System.out.println(word);
        }
    }
}

```

Output:

Output

```
Length of str1: 13
Concatenated String: Hello World - Java Programming
Character at index 1 in str2: a
Substring of str2 (5 to 16): Programming
Index of 'World' in str1: 7
str1 equals str2? false
str1 equalsIgnoreCase 'hello world'? true
str2 in uppercase: JAVA PROGRAMMING
str2 in lowercase: java programming
str1 trimmed: 'Hello World'
str2 with 'Java' replaced by 'Python': Python Programming
Words in str2:
Java
Programming

=== Code Execution Successful ===
```