

JAVA LAB

BCA- DS - 452A

Manav Rachna International Institute of Research and Studies

**School of Computer Applications
Department of Computer Applications**

Submitted By	
Student Name	Alance Tanwar
Roll No	23/SCA/BCA(AI&ML)/03
Programme	Bachelor of Computer Applications
Semester	4 th Semester
Section/Group	4C/A
Department	Computer Applications
Batch	2023-26
Submitted To	
Faculty Name	Dr. Priyanka Sharma.



SCHOOL OF COMPUTER APPLICATIONS

LAB SESSION 1

Write java program to print hello world

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

Create a java program to check whether a number entered by user is even or odd
Create a java program to print the average and sum of 5 numbers entered by user.
Program to calculate the factorial of a number
Program to print Fibonacci series up to n terms

LAB SESSION 2

Program to reverse a number
Program to check if a number is a palindrome
Program for a simple calculator
Program to check if a number is prime
Program to check if a number is an Armstrong number
Find the Largest of Three Numbers using ternary operator

LAB SESSION 3

Print Multiplication Table
Calculate Sum and Average of Array Elements
Reverse a String
Find Factorial of a Number Using Recursion
Sort an Array in Ascending Order

LAB SESSION 4

Check Palindrome for a String
Count Vowels and Consonants in a String
Write a program to demonstrate type casting.
Write a program to generate prime numbers between 1 & given number

LAB SESSION 5

Program to Demonstrate a Simple Class with Methods
Program for Class with Parameterized Constructor
Program to Find the Area of a Rectangle Using Methods
Program for Bank Account Class with Deposit and Withdraw Methods
Program to Demonstrate Method Overloading

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.

Write a program to count the number of objects created for a class using static member function

LAB SESSION 7

Write a program to design a class using abstract methods and abstract classes.

Write a program to demonstrate the use of multilevel inheritance

Write a program to demonstrate the use of multiple inheritance

Write a program that show the partial implementation of Interface

Write a program to design a string class that perform string method(Equal, Reverse the string, change case).

Write a program to handle the exception using try and multiple catch block.

Write a program to create a package that access the member of External class as well as same package.

Write a program that import the user define package and access the Member variable of classes that contained by package.

Write a program to handle the user defined exception using throw keyword.

LAB SESSION 8

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

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

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

Write a program to create a class component that shows controls and event handling on that controls.

Write a program to draw the line, Rectangle, oval, text using the graphics method.

Write a program to create a menu using the frame.

Write a program to create a dialogbox.

Write a program to implement the flow layout and border layout.

Write a program to implement the gridLayout, cardLayout.

Write a program to create Frame that display the student information

Lab Session 1 by Chaitanya Khanna

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
```

3. 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
```

4. 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
```

5. 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
```

6. 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
```

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
```

Written by Chaitanya Khanna

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.
```

Written by Chaitanya Khanna

3. 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
```

Written by Chaitanya Khanna

4. 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.
```

5. 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.
```

Written by Chaitanya Khanna

6. 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
```

Written by Chaitanya Khanna

Lab Session 3 by Chaitanya Khanna

1. Program to Print Multiplication Table

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 number = scanner.nextInt();

        System.out.println("Multiplication Table for " + number + ":");
        for (int i = 1; i <= 10; i++) {
            System.out.println(number + " x " + i + " = " + (number * i));
        }

        scanner.close();
    }
}
```

Output:

```
Enter a number: 15
Multiplication Table for 15:
15 x 1 = 15
15 x 2 = 30
15 x 3 = 45
15 x 4 = 60
15 x 5 = 75
15 x 6 = 90
15 x 7 = 105
15 x 8 = 120
15 x 9 = 135
15 x 10 = 150
```

Written by Chaitanya Khanna

2. Java Program to Calculate Sum and Average of Array Elements Input:

```
import java.util.Scanner;

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

        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();

        int[] numbers = new int[n];
        int sum = 0;

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            numbers[i] = scanner.nextInt();
            sum += numbers[i];
        }

        double average = (double) sum / n;

        System.out.println("Sum: " + sum);
        System.out.println("Average: " + average);

        scanner.close();
    }
}
```

Output:

```
Enter the number of elements: 3
Enter the elements:
12
21
11
Sum: 44
Average: 14.66666666666666
```

Written by Chaitanya Khanna

3. Create a java Program to Reverse a String 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 string: ");
        String input = scanner.nextLine();

        String reversed = "";
        for (int i = input.length() - 1; i >= 0; i--) {
            reversed += input.charAt(i);
        }

        System.out.println("Reversed string: " + reversed);

        scanner.close();
    }
}
```

Output:

```
Enter a string: 153
Reversed string: 351
```

Written by Chaitanya Khanna

4. Create a java Program to Find Factorial of a Number Using Recursion 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 number = scanner.nextInt();

        System.out.println("Factorial of " + number + " is: " + factorial(number));

        scanner.close();
    }

    public static int factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        }
        return n * factorial(n - 1);
    }
}
```

Output:

```
Enter a number: 14
Factorial of 14 is: 1278945280
```

Written by Chaitanya Khanna

5. Program to Sort an Array in Ascending Order Input:

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

        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();

        int[] numbers = new int[n];

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            numbers[i] = scanner.nextInt();
        }

        Arrays.sort(numbers);
        System.out.println("Sorted array in ascending order: ");
        for (int num : numbers) {
            System.out.print(num + " ");
        }

        scanner.close();
    }
}
```

Output:

```
Enter the number of elements: 5
Enter the elements:
10
9
8
7
6
Sorted array in ascending order:
6 7 8 9 10
```

Written by Chaitanya Khanna

Lab Session 4 by Chaitanya Khanna

1. Program to Check Palindrome for a String 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 string: ");
        String original = scanner.nextLine();
        scanner.close();

        String reversed = "";
        for (int i = original.length() - 1; i >= 0; i--) {
            reversed += original.charAt(i);
        }

        if (original.equalsIgnoreCase(reversed)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }
}
```

- Output:

```
Enter a string: 12
The string is not a palindrome.
```

Written by Chaitanya Khanna

2. Java Program to Count Vowels and Consonants in a String 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 string: ");
        String input = scanner.nextLine();
        scanner.close();

        int vowels = 0, consonants = 0;
        String str = input.toLowerCase();

        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (ch >= 'a' && ch <= 'z') {
                if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
                    {
                        vowels++;
                    } else {
                        consonants++;
                    }
            }
        }
        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
    }
}
```

Run

Output:

```
Enter a string: Atul Vashist
Vowels: 4
Consonants: 7
```

Written by Chaitanya Khanna

3. Write a program to demonstrate type casting. Input:

```
import java.util.Scanner;

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

        System.out.print("Enter an integer: ");
        int intValue = scanner.nextInt();

        double doubleValue = intValue;
        System.out.println("Implicit Casting (int to double): " + doubleValue);

        System.out.print("Enter a double value: ");
        double inputDouble = scanner.nextDouble();

        int castedInt = (int) inputDouble;
        System.out.println("Explicit Casting (double to int): " + castedInt);

        scanner.close();
    }
}
```

Output:

```
Enter an integer: 15
Implicit Casting (int to double): 15.0
Enter a double value: 30
Explicit Casting (double to int): 30
```

Written by Chaitanya Khanna

4. Write a program to generate prime numbers between 1 & given 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();
        scanner.close();
        System.out.println("Prime numbers between 1 and " + num + ":");
        for (int i = 2; i <= num; i++) {
            if (isPrime(i)) {
                System.out.print(i + " ");
            }
        }
    }
    public static boolean isPrime(int n) {
        if (n < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

Output:

```
Enter a number: 559
Prime numbers between 1 and 559:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113
127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233
239 241 251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359
367 373 379 383 389 397 401 409 419 421 431 433 439 443 449 457 461 463 467 479 487
491 499 503 509 521 523 541 547 557
== Code Execution Successful ==
```

Lab Session 5 by Chaitanya Khanna

1. Program to Demonstrate a Simple Class with Methods

Input:

```
import java.util.Scanner;
class SimpleClass {
    private String message;
    public void setMessage(String msg) {
        message = msg;
    }
    public void displayMessage() {
        System.out.println("Message: " + message);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a message: ");
        String inputMessage = scanner.nextLine();
        scanner.close();

        SimpleClass obj = new SimpleClass();
        obj.setMessage(inputMessage);
        obj.displayMessage();
    }
}
```

Output:

```
Enter a message: Hello From Atul
Message: Hello From Atul
```

2. Java Program for Class with Parameterized Constructor

Input:

```
import java.util.Scanner;
class Person {
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}

public class Main{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter name: ");
        String name = scanner.nextLine();
        System.out.print("Enter age: ");
        int age = scanner.nextInt();
        scanner.close();
        Person person = new Person(name, age);
        person.displayInfo();
    }
}
```

Output:

```
Enter name: Atul Vashist
Enter age: 19
Name: Atul Vashist
Age: 19
```

3. Write a Program to Find the Area of a Rectangle Using Methods Input:

```
import java.util.Scanner;
class Rectangle {
    private double length;
    private double width;
    public void setDimensions(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public double calculateArea() {
        return length * width;
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter length of the rectangle: ");
        double length = scanner.nextDouble();
        System.out.print("Enter width of the rectangle: ");
        double width = scanner.nextDouble();
        scanner.close();
        Rectangle rect = new Rectangle();
        rect.setDimensions(length, width);

        System.out.println("Area of the rectangle: " + rect.calculateArea());
    }
}
```

Run

Output:

```
Enter length of the rectangle: 24
Enter width of the rectangle: 15
Area of the rectangle: 360.0
```

4. Write a Program for Bank Account Class with Deposit and Withdraw Methods

Input:

```
import java.util.Scanner;
class BankAccount {
    private double balance;
    public BankAccount(double initialBalance) {
        this.balance = initialBalance;
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance
                .");
        }
    }
}
```

```
    public void displayBalance() {
        System.out.println("Current Balance: " + balance);
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter initial balance: ");
        double initialBalance = scanner.nextDouble();

        BankAccount account = new BankAccount(initialBalance);

        System.out.print("Enter deposit amount: ");
        double depositAmount = scanner.nextDouble();
        account.deposit(depositAmount);

        System.out.print("Enter withdrawal amount: ");
        double withdrawAmount = scanner.nextDouble();
        account.withdraw(withdrawAmount);

        account.displayBalance();

        scanner.close();
    }
}
```

Output:

```
Enter initial balance: 200000.
Enter deposit amount: 1500000
Deposited: 1500000.0
Enter withdrawal amount: 20000
Withdrawn: 20000.0
Current Balance: 1680000.0
```

5. Write Program to Demonstrate Method Overloading

Input:

```
class MathOperations {  
    public int add(int a, int b) {  
        return a + b;  
    }  
    //Written by ATUL VASHIST  
    public double add(double a, double b) {  
        return a + b;  
    }  
    //Written by ATUL VASHIST  
    public int add(int a, int b, int c) {  
        return a + b + c;  
    }  
}  
//Written by ATUL VASHIST  
public class Main {  
    public static void main(String[] args) {  
        MathOperations math = new MathOperations();  
  
        System.out.println("Sum of two integers: " + math.add(5, 10));  
        System.out.println("Sum of two doubles: " + math.add(5.5, 10.5));  
        System.out.println("Sum of three integers: " + math.add(5, 10, 15));  
        //Written by ATUL VASHIST  
    }  
}
```

Output:

```
Sum of two integers: 15  
Sum of two doubles: 16.0  
Sum of three integers: 30  
  
== Code Execution Successful ==
```

Lab Session 6 by Chaitanya Khanna

1. Program to Demonstrate Static Methods Input:

```
class Main {  
    static void displayMessage() {  
        System.out.println("Hello! This is Atul Vashist");  
    }  
  
    public static void main(String[] args) {  
        displayMessage();  
    }  
}
```

Output:

```
Hello! This is Atul Vashist  
==== Code Execution Successful ===
```

2. Java Program to Demonstrate Method Overriding

Input:

```
class Animal {
    void makeSound() {
        System.out.println("Animal makes a sound.");
    }
}

class Dog extends Animal {
    @Override
    void makeSound() {
        System.out.println("Dog barks.");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal animal = new Animal();
        animal.makeSound();

        Animal dog = new Dog();
        dog.makeSound();
    }
}
```

Output:

```
Animal makes a sound.
Dog barks.

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

3. Write a Program to Demonstrate Getters and Setters

Input:

```
class Bike {  
    private String name;  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Bike bike = new Bike();  
        bike.setName("BMW 1250"); // Written by Atul Vashist.  
        System.out.println("Bike name: " + bike.getName());  
    }  
}
```

Output:

```
Bike name: BMW 1250
```

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

4. Write a Program to Demonstrate a Class with Multiple

Methods Input:

```
public class Calculator {  
    public int add(int a, int b) {  
        return a + b;  
    }  
    public int subtract(int a, int b) {  
        return a - b;  
    }  
    public int multiply(int a, int b) {  
        return a * b;  
    }  
    public int divide(int a, int b) {  
        if (b != 0) {  
            return a / b;  
        } else {  
            System.out.println("Error! Division by zero.");  
            return 0;  
        }  
    }  
    public static void main(String[] args) {  
        Calculator calc = new Calculator();  
        System.out.println("Addition: " + calc.add(10, 5));  
        System.out.println("Subtraction: " + calc.subtract(10, 5));  
        System.out.println("Multiplication: " + calc.multiply(10, 5));  
        System.out.println("Division: " + calc.divide(10, 5));  
        System.out.println("Division by zero: " + calc.divide(10, 0));  
    }  
}
```

Output:

```
Addition: 15  
Subtraction: 5  
Multiplication: 50  
Division: 2  
Error! Division by zero.  
Division by zero: 0  
  
== Code Execution Successful ==
```

5. Write a Program to Demonstrate Object Passing in Methods Input:

```
class Student {  
    String name;  
    int age;  
    Student(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
    void display() {  
        System.out.println("Name: " + name + ", Age: " + age);  
    }  
}  
class Demo {  
    void updateStudent(Student s) {  
        s.name = "Updated " + s.name;  
        s.age += 1;  
    }  
}  
public class Main {  
    public static void main(String[] args) {  
        Student student = new Student("Atul", 20);  
        Demo demo = new Demo();  
        System.out.println("Before update:");  
        student.display();  
        demo.updateStudent(student);  
        System.out.println("After update:");  
        student.display();  
    }  
}
```

Output:

```
Before update:  
Name: Atul, Age: 20  
After update:  
Name: Updated Atul, Age: 21  
==== Code Execution Successful ===
```

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

Input:

```
class Shape {  
    double length, width;  
    Shape(double length, double width) {  
        this.length = length;  
        this.width = width;  
    }  
}  
class Rectangle extends Shape {  
    Rectangle(double length, double width) {  
        super(length, width);  
    }  
    double getArea() {  
        return length * width;  
    }  
    double getPerimeter() {  
        return 2 * (length + width);  
    }  
}  
public class Main {  
    public static void main(String[] args) {  
        Rectangle rectangle = new Rectangle(5.0, 3.0);  
        System.out.println("Area: " + rectangle.getArea());  
        System.out.println("Perimeter: " + rectangle.getPerimeter());  
    }  
}
```

Output:

```
Area: 15.0  
Perimeter: 16.0  
==== Code Execution Successful ===
```

7. Write a program to count the number of objects created for a class using static member function

Input:

```
class ObjectCounter {  
    private static int count = 0;  
  
    ObjectCounter() {  
        count++;  
    }  
  
    static int getCount() {  
        return count;  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        ObjectCounter obj1 = new ObjectCounter();  
        ObjectCounter obj2 = new ObjectCounter();  
        ObjectCounter obj3 = new ObjectCounter();  
  
        System.out.println("Number of objects created: " + ObjectCounter  
                           .getCount());  
    }  
}
```

Output:

```
Number of objects created: 3  
== Code Execution Successful ==
```

Lab Session 7 by Chaitanya Khanna

1. Write a program to design a class using abstract methods and abstract classes.

Input:

```
abstract class Shape {  
    abstract void draw();  
    void display() {  
        System.out.println("This is a shape.");  
    }  
}  
class Circle extends Shape {  
    @Override  
    void draw() {  
        System.out.println("Drawing a Circle.");  
    }  
}  
class Rectangle extends Shape {  
    @Override  
    void draw() {  
        System.out.println("Drawing a Rectangle.");  
    }  
}  
public class Main {  
    public static void main(String[] args) {  
        Shape circle = new Circle();  
        circle.display();  
        circle.draw();  
  
        Shape rectangle = new Rectangle();  
        rectangle.display();  
        rectangle.draw();  
    }  
}
```

Output:

```
This is a shape.  
Drawing a Circle.  
This is a shape.  
Drawing a Rectangle.
```

2. Java Program to demonstrate the use of multilevel inheritance Input:

```
class Vehicle {  
    void start() {  
        System.out.println("Vehicle is starting.");  
    }  
}  
  
class Bike extends Vehicle {  
    void accelerate() {  
        System.out.println("Bike is accelerating.");  
    }  
}  
  
class SportsBike extends Bike {  
    void turbo() {  
        System.out.println("SportsBike turbo mode activated!");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        SportsBike myBike = new SportsBike();  
        myBike.start();  
        myBike.accelerate();  
        myBike.turbo();  
    }  
}
```

Output:

```
Vehicle is starting.  
Bike is accelerating.  
SportsBike turbo mode activated!  
  
== Code Execution Successful ==
```

3. Write a Program to demonstrate the use of multiple inheritance

Input:

```
interface Engine {
    void startEngine();
}

interface Wheels {
    void rotateWheels();
}

class Bike implements Engine, Wheels {
    public void startEngine() {
        System.out.println("Bike engine is starting.");
    }

    public void rotateWheels() {
        System.out.println("Bike wheels are rotating.");
    }
}

public class Main {
    public static void main(String[] args) {
        Bike myBike = new Bike();
        myBike.startEngine();
        myBike.rotateWheels();
    }
}
```

Output:

```
Bike engine is starting.
Bike wheels are rotating.

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

4. Write a Program that show the partial implementation of Interface

Input:

```
interface Engine {  
    void startEngine();  
    void stopEngine();  
}  
  
abstract class Bike implements Engine {  
    public void startEngine() {  
        System.out.println("Bike engine is starting.");  
    }  
}  
  
class SportsBike extends Bike {  
    public void stopEngine() {  
        System.out.println("Bike engine is stopping.");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        SportsBike myBike = new SportsBike();  
        myBike.startEngine();  
        myBike.stopEngine();  
    }  
}
```

Output:

```
Bike engine is starting.  
Bike engine is stopping.  
  
*** Code Execution Successful ***
```

5. Write a Program to design a string class that perform string method(Equal, Reverse the string, change case).

Input:

```
- class MyString {
    private String str;

    - public MyString(String str) {
        this.str = str;
    }
    - public boolean isEqual(String otherStr) {
        return str.equals(otherStr);
    }
    - public String reverse() {
        return new StringBuilder(str).reverse().toString();
    }
    - public String changeCase() {
        StringBuilder result = new StringBuilder();
        for (char ch : str.toCharArray()) {
            if (Character.isUpperCase(ch)) {
                result.append(Character.toLowerCase(ch));
            } else {
                result.append(Character.toUpperCase(ch));
            }
        }
        return result.toString();
    }
}
public class Main {
    public static void main(String[] args) {
        MyString myStr = new MyString("Hello World");
        System.out.println("Original String: " + myStr);
        System.out.println("Reversed String: " + myStr.reverse());
        System.out.println("Case Changed String: " + myStr.changeCase());
        System.out.println("Is Equal to 'Hello World': " + myStr.isEqual("Hello World"));
    }
}
```

Output:

```
Original String: MyString@2a139a55
Reversed String: dlrow olleH
Case Changed String: hELLO wORLD
Is Equal to 'Hello World': true
```

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

6. Write a program to handle the exception using try and multiple catch blocks.

Input:

```
class MyString {  
    private String str;  
  
    public MyString(String str) {  
        this.str = str;  
    }  
    public boolean isEqual(String otherStr) {  
        try {  
            return str.equals(otherStr);  
        } catch (NullPointerException e) {  
            System.out.println("Error: Null value encountered");  
            return false;  
        }  
    }  
    public String reverse() {  
        try {  
            return new StringBuilder(str).reverse().toString();  
        } catch (NullPointerException e) {  
            System.out.println("Error: Null value encountered");  
            return "";  
        }  
    }  
    public String changeCase() {  
        try {  
            StringBuilder result = new StringBuilder();  
            for (char ch : str.toCharArray()) {  
                if (Character.isUpperCase(ch)) {  
                    result.append(Character.toLowerCase(ch));  
                } else {  
                    result.append(Character.toUpperCase(ch));  
                }  
            }  
            return result.toString();  
        } catch (NullPointerException e) {  
            System.out.println("Error: Null value encountered");  
            return "";  
        }  
    }  
}  
public class Main {  
    public static void main(String[] args) {  
        try {  
            MyString myStr = new MyString("Hello World");  
            System.out.println("Original String: " + myStr);  
            System.out.println("Reversed String: " + myStr.reverse());  
            System.out.println("Case Changed String: " + myStr.changeCase());  
            System.out.println("Is Equal to 'Hello World': " + myStr.isEqual("Hello World"));  
            MyString nullStr = new MyString(null);  
            System.out.println("Reversed Null String: " + nullStr.reverse());  
        } catch (Exception e) {  
            System.out.println("An unexpected error occurred: " + e.getMessage());  
        }  
    }  
}
```

Output:

```
Original String: MyString@2a139a55  
Reversed String: dlroW olleH  
Case Changed String: hELLO wORLD  
Is Equal to 'Hello World': true  
ERROR!  
Error: Null value encountered  
Reversed Null String:
```

7. Write a program

to create a package that access the member of External class
as well as same package

Input:

```
public class MainApp {  
    static class ExternalClass {  
        public void showExternal() {  
            System.out.println("This is the External Class!");  
        }  
    }  
    static class InternalClass {  
        public void showInternal() {  
            System.out.println("This is the Internal Class!");  
        }  
  
        public void accessExternal() {  
            ExternalClass external = new ExternalClass();  
            external.showExternal();  
        }  
    }  
    public static void main(String[] args) {  
        ExternalClass external = new ExternalClass();  
        external.showExternal();  
        InternalClass internal = new InternalClass();  
        internal.showInternal();  
        internal.accessExternal();  
    }  
}
```

Output:

```
This is the External Class!  
This is the Internal Class!  
This is the External Class!  
  
==== Code Execution Successful ===
```

8. Write a program

that import the user-defined package and accesses the Member variable of classes that contained by the package.

Input:

```
class ExternalClass {  
    public String externalMessage = "Hello from ExternalClass!";  
}  
class MyClass {  
    public String message = "Hello from MyClass!";  
}  
  
public class Main {  
    public static void main(String[] args) {  
  
        MyClass myObj = new MyClass();  
        System.out.println(myObj.message);  
  
        ExternalClass extObj = new ExternalClass();  
        System.out.println(extObj.externalMessage);  
    }  
}
```

Output:

```
Hello from MyClass!  
Hello from ExternalClass!  
  
== Code Execution Successful ==
```

9. Write a program

to handle the user-defined exception using the throw keyword.

Input:

```
class InvalidAgeException extends Exception {  
    public InvalidAgeException(String message) {  
        super(message);  
    }  
}  
class Voter {  
    private int age;  
    public Voter(int age) throws InvalidAgeException {  
        if (age < 18) {  
            throw new InvalidAgeException("Age must be 18 or above to vote.");  
        }  
        this.age = age;  
    }  
    public void vote() {  
        System.out.println("You are eligible to vote.");  
    }  
}  
public class Main {  
    public static void main(String[] args) {  
        try {  
            Voter voter1 = new Voter(16);  
            voter1.vote();  
        } catch (InvalidAgeException e) {  
            System.out.println("Exception Caught: " + e.getMessage());  
        }  
        try {  
            Voter voter2 = new Voter(20);  
            voter2.vote();  
        } catch (InvalidAgeException e) {  
            System.out.println("Exception Caught: " + e.getMessage());  
        }  
    }  
}
```

Exception Caught: Age must be 18 or above to vote.
You are eligible to vote.

==== Code Execution Successful ===

Lab Session 8 by Chaitanya Khanna

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

```
public class BikeStringExample {  
    public static void main(String[] args) {  
        String bikeInfo = "My bike is a Royal Enfield Classic 350";  
  
        String sub1 = bikeInfo.substring(11);  
        System.out.println("Substring from index 11: " + sub1);  
  
        String sub2 = bikeInfo.substring(11, 26);  
        System.out.println("Substring from index 11 to 25: " + sub2);  
  
        String replacedText = bikeInfo.replace("Royal Enfield", "Yamaha R15");  
        System.out.println("After replace: " + replacedText);  
  
        String[] words = bikeInfo.split(" ");  
        System.out.println("Split by space:");  
        for (String word : words) {  
            System.out.println(word);  
        }  
    }  
}
```

Output:

```
Substring from index 11: a Royal Enfield Classic 350  
Substring from index 11 to 25: a Royal Enfield  
After replace: My bike is a Yamaha R15 Classic 350  
Split by space:  
My  
bike  
is  
a  
Royal  
Enfield  
Classic
```

2. Create a custom exception, Age Exception, that checks if a person's age is valid (above 18). In Java Input:

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

public class Main {
    public static void checkAge(int age) throws AgeException {
        if (age < 18) {
            throw new AgeException("Age is not valid. You must be 18 or older.");
        } else {
            System.out.println("Age is valid. You are allowed.");
        }
    }

    public static void main(String[] args) {
        int age = 16;

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

Output:

```
Exception: Age is not valid. You must be 18 or older.
```

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

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";  
        String str2 = "Atul Vashist! ";  
        String fullStr = str1 + " " + str2;  
  
        System.out.println("Original String: '" + fullStr + "'");  
  
        System.out.println("Length: " + fullStr.length());  
  
        String concatStr = str1.concat(" ").concat(str2);  
        System.out.println("Concatenated: " + concatStr);  
  
        char ch = fullStr.charAt(2);  
        System.out.println("Character at index 2: " + ch);  
  
        String sub = fullStr.substring(2, 7);  
        System.out.println("Substring from index 2 to 6: " + sub);  
  
        int index = fullStr.indexOf("Atul");  
        System.out.println("Index of 'Atul': " + index);  
    }  
}
```

```
boolean isEqual = str2.trim().equals("Atul Vashist!");
System.out.println("Is str2 equal to 'Atul Vashist!'? " + isEqual);

System.out.println("Uppercase: " + fullStr.toUpperCase());
System.out.println("Lowercase: " + fullStr.toLowerCase());

String trimmed = fullStr.trim();
System.out.println("Trimmed string: '" + trimmed + "'");

String replaced = fullStr.replace("Atul", "Captain");
System.out.println("After replace: " + replaced);

String[] parts = fullStr.trim().split(" ");
System.out.println("Split by space:");
for (String part : parts) {
    System.out.println(part);
}
}
```

Output:

```
Original String: ' Hello Atul Vashist!
Length: 23
Concatenated: Hello Atul Vashist!
Character at index 2: H
Substring from index 2 to 6: Hello
Index of 'Atul': 8
Is str2 equal to 'Atul Vashist!'? true
Uppercase: HELLO ATUL VASHIST!
Lowercase: hello atul vashist!
Trimmed string: 'Hello Atul Vashist!'
After replace: Hello Captain Vashist!
Split by space:
Hello
Atul
Vashist!

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

4. Write a program to create a class component that shows controls and event handling on those controls.

Input:

```
import java.util.Scanner;

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

        System.out.print("Enter your name: ");
        String name = scanner.nextLine();

        System.out.print("Subscribe to newsletter? (yes/no): ");
        String subscribeInput = scanner.nextLine();
        boolean isSubscribed = subscribeInput.equalsIgnoreCase("yes");

        System.out.println("\n--- Output ---");
        System.out.println("Name: " + name);
        System.out.println("Subscribed: " + (isSubscribed ? "Yes" : "No"));

        scanner.close();
    }
}
```

Output:

```
Enter your name: Atul Vashist
Subscribe to newsletter? (yes/no): yes
```

```
--- Output ---
```

```
Name: Atul Vashist
```

```
Subscribed: Yes
```

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

5. Write a program to draw a line, a Rectangle, an oval, and text using the graphics method. Input:

```
public class GraphicsSimulation {  
    public static void main(String[] args) {  
        System.out.println("Simulating Drawing using Console:");  
        System.out.println("-----");  
        System.out.println("Drawing a Line from (20, 20) to (200, 20)");  
        System.out.println("Drawing a Rectangle at (20, 40) with width=100 and  
                           height=60");  
        System.out.println("Drawing an Oval at (140, 40) with width=100 and height  
                           =60");  
        System.out.println("Displaying Text at (20, 130): \"Hello, Graphics!\"");  
    }  
}
```

Output:

```
Simulating Drawing using Console:  
-----  
Drawing a Line from (20, 20) to (200, 20)  
Drawing a Rectangle at (20, 40) with width=100 and height=60  
Drawing an Oval at (140, 40) with width=100 and height=60  
Displaying Text at (20, 130): "Hello, Graphics!"  
  
==== Code Execution Successful ===
```

6. Write a program to create a menu using the frame. Input:

```
import java.util.Scanner;
public class ConsoleMenuExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;
        do {
            System.out.println("== Menu ==");
            System.out.println("1. Open");
            System.out.println("2. Save");
            System.out.println("3. Exit");
            System.out.print("Choose an option: ");
            choice = scanner.nextInt();
            switch (choice) {
```

```
        case 1:
            System.out.println("Opening file...");
            break;
        case 2:
            System.out.println("Saving file...");
            break;
        case 3:
            System.out.println("Exiting...");
            break;
        default:
            System.out.println("Invalid choice.");
    }
} while (choice != 3);
scanner.close();
}
}
```

Output:

```
==== Menu ====
1. Open
2. Save
3. Exit
Choose an option: 1
Opening file...
==== Menu ====
1. Open
2. Save
3. Exit
Choose an option: 2
Saving file...
==== Menu ====
1. Open
2. Save
3. Exit
Choose an option: 3
Exiting...

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

7. Write a program to create a dialog box.

Input:

```
public class ConsoleDialogSim {
    public static void main(String[] args) {
        System.out.println(">>> [Dialog Box]");
        System.out.println("This is a dialog box!");
    }
}
```

Output:

```
>>> [Dialog Box]
This is a dialog box!

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

8. Write a program to implement the flow layout and border layout.

Input:

```
public class ConsoleLayoutSim {  
    public static void main(String[] args) {  
        System.out.println("Simulating Layouts:");  
        System.out.println("FlowLayout Panel: [One] [Two] [Three] [Four] [Five]");  
        System.out.println("BorderLayout Panel:");  
        System.out.println("    North: [North]");  
        System.out.println("    West: [West]    Center: [Center]    East: [East]");  
        System.out.println("    South: [South]");  
    }  
}
```

Output:

```
Simulating Layouts:  
FlowLayout Panel: [One] [Two] [Three] [Four] [Five]  
BorderLayout Panel:  
    North: [North]  
    West: [West]    Center: [Center]    East: [East]  
    South: [South]  
==== Code Execution Successful ===
```

9. Write a program to implement the gridLayout, cardLayout

Input:

```
public class ConsoleLayoutSim {  
    public static void main(String[] args) {  
        System.out.println("GridLayout Simulation (2x2):");  
        System.out.println("[Grid 1] [Grid 2]");  
        System.out.println("[Grid 3] [Grid 4]");  
  
        System.out.println("\nCardLayout Simulation:");  
        System.out.println("Showing Card 1: 'This is Card 1'");  
        System.out.println("Press Enter to switch to Card 2...");  
        new java.util.Scanner(System.in).nextLine();  
        System.out.println("Now showing Card 2: 'This is Card 2'");  
    }  
}
```

Output:

```
GridLayout Simulation (2x2):  
[Grid 1] [Grid 2]  
[Grid 3] [Grid 4]  
  
CardLayout Simulation:  
Showing Card 1: 'This is Card 1'  
Press Enter to switch to Card 2...  
  
Now showing Card 2: 'This is Card 2'  
  
==== Code Execution Successful ===
```

10. Write a program to create Frame that display the student information

Input:

```
public class ConsoleStudentInfo {  
    public static void main(String[] args) {  
        System.out.println("Student Information:");  
        System.out.println("-----");  
        System.out.println("Name: Atul Vashist");  
        System.out.println("Roll No: 009");|  
        System.out.println("Course: BCA (AIML)");  
        System.out.println("College: Manav Rachna");  
    }  
}
```

Output:

```
Student Information:  
-----  
Name: Atul Vashist  
Roll No: 009  
Course: BCA (AIML)  
College: Manav Rachna  
  
==== Code Execution Successful ===
```