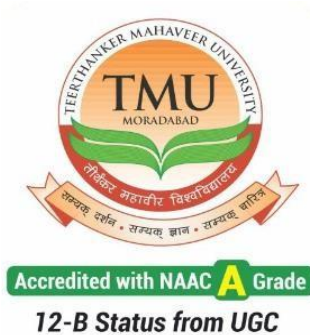


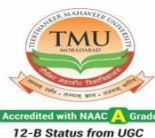
LAB MANUAL
JAVA PROGRAMMING
LAB (ECS-456)

IV Semester

FACULTY OF ENGINEERING



TEERTHANKER MAHAVEER UNIVERSITY
MORADABAD
2023-24



FACULTY OF ENGINEERING

(TEERTHANKER MAHAVEER UNIVERSITY MORADABAD)

LAB MANUAL

Submitted By

Student Name:.....

RollNo :.....

Branch:.....**Section**.....

Year**Semester**.....

Submitted To

FACULTY INCHARGE

OBJECTIVES:

1. To introduce students to the basic knowledge of programming fundamentals of java language.
2. To impart writing skill of java programming to the students and solving problems.
3. To impart the concepts like looping, array, functions, class, interface .

COURSE OUTCOME:

Course Outcomes:	On completion of the course, the students will be :	BTL
CO1.	Applying knowledge to solve real world problems based on object-oriented principles..	3
CO2.	Applying the basic approach of graphical user interface design using Abstract window toolkit, Applet and swing packages, create some application that are based upon some real world scenario	3,4
CO3.	Analyzing the concept of database handling and creating application that are able to communicate with various database.	3,4
CO4.	Analyzing the web architecture for creating applications using servlets and java server pages	3,4
CO5.	Analyzing the Client server architecture, Understanding the Remote method invocation architecture and creating basic application using Remote method invocation.	3,4

Program Outcomes for Engineering

PO – 1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO – 2	Problem analysis& Solving: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO – 3	Design/development of solutions: Design solutions for complex engineering problems and design system components the public health and safety, and the cultural, societal, and environmental considerations or processes that meet the specified needs with appropriate consideration for
PO – 4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO – 5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO – 6	Social Interaction & effective citizenship: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO – 7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO – 8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO – 9	Attitude (Individual and team work): Function effectively as an individual, and as member or leader in diverse teams, and in multidisciplinary settings.
PO – 10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clean instructions.
PO – 11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO – 12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes (PSOs)

PSO – 1	Understanding the knowledge of basic sciences, humanities and technical management courses of the program. Able to solve engineering problems of real time projects in the field of computer science and information technology.
PSO – 2	Understanding the phases of software project development life cycle and various roles.
PSO – 3	Applying hardware and software skills pertinent to practices in the field of computer science and information technology while acquiring mathematical foundations, algorithmic principles along with proper judgment through projects and industrial interactions.
PSO – 4	Analyzing the various storage structures of data on different platforms along with security issues.
PSO - 5	Analyzing large data samples and discovers knowledge to provide solution to engineering problem.
PSO - 6	Developing skills of practical competency with emerging technologies, programming languages and open source platforms.
PSO - 7	Developing effective artificial intelligence based solutions for real world problem.

PSO No	Program Objective	Course Outcome	Relevance	Justification
PO1	Engineering Knowledge	CO1	3(Strong)	Strong use of engineering fundamentals to build modular, reusable and scalable software using object-oriented design.
		CO2	3(Strong)	Applies technical skills to build user-facing applications in Java.
		CO3	3(Strong)	Requires solid engineering and computing skills to create and connect databases
		CO4	3(Strong)	Requires core technical skills to develop web applications and understand request/response flow.
		CO5	3(Strong)	Applies deep technical knowledge in building distributed applications.
PO2	Problem Analysis	CO1	2(Medium)	Requires identifying real-world problems and analyzing them into class-based structures.
		CO2	2(Medium)	GUI layout and component selection require thoughtful problem analysis.
		CO3	3(Strong)	Understanding and analyzing database schemas and logic is crucial.
		CO4	3(Strong)	Student needs to analyze user requirements and server behavior to structure servlet-based apps.
		CO5	3(Strong)	Analyzing request/response and remote communication structures is essential.
PO3	Design/Development of solutions	CO1	3(Strong)	Involves designing classes, methods, inheritance hierarchies – core solution development tasks
		CO2	3(Strong)	Designing responsive interfaces is a clear instance of development and solution design
		CO3	3(Strong)	Database-driven applications need careful planning and development.
		CO4	3(Strong)	Designing web apps that are modular, secure, and maintainable is part of complex solution development.
		CO5	3(Strong)	RMI applications require careful design of stubs, registry, and objects

PO4	Conduct Investigations of Complex problems	CO1	1(Low)	Includes basic code testing and debugging, contributing modestly to investigations.
		CO2		

Lab Equipment

Following hardware and software are necessary to perform the experiments in java Programming Lab.

Hardware:

1. A computer that can execute Java programs.

Software:

1. Windows Operating System.
2. Compilers: JDK, Net Beans

Programming Languages:

- 1 java

Evaluation of practical course

Practical Courses Internal Evaluation (50 marks)

Experiment (30 Marks)	Attendance (10 Marks)	Viva (10 Marks)	Total Internal (50 Marks)
--------------------------	--------------------------	--------------------	------------------------------

Practical Courses External Evaluation (50 marks)

The external evaluation would also be done by the External Examiner based on the experiment performed during the examination.

Experiment (30 Marks)	File Work (10 Marks)	Viva (10 Marks)	Total External (50 Marks)
--------------------------	-------------------------	--------------------	------------------------------

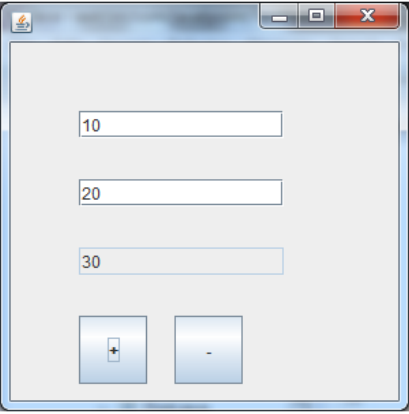
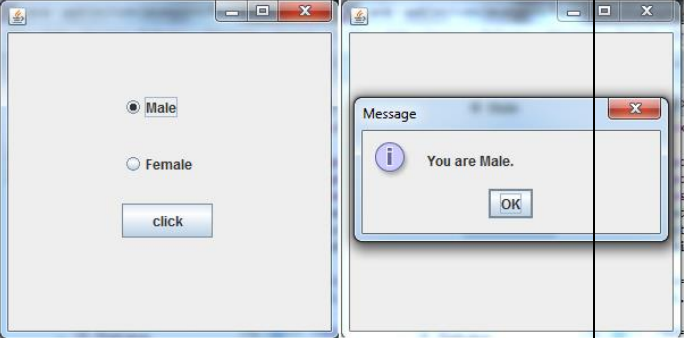
INDEX

Lab Exercises

S.No.	Exercise No.	Program No.	Page No.	Signature of faculty
1	Basic knowledge	Write a program to print Hello JAVA.		
		Write a program to print whatever value is given by user using command line arguments.		
		Write a program print the sum of two numbers given by user using command line argument.		
		Write a program print the sum of two numbers given by user using Scanner Class.		
		Write a program to calculate sum and average of five numbers entered by user.		
		a. Write a program to interchange values of two numbers using third variable.		
		b. Write a program to interchange values of two numbers without using third variable.		
		Write a program to find and display greater number from three numbers.		
		Write a program to find whether a year is leap or not.		
		Write a program to find factorial of a given number.		
2	Basic Data Types knowledge	Write a program to Generate Fibonacci series as per given range which is entered by the user.		
		Write a Program to print the reverse of an integer number entered by user.		
		Write a program to find the sum of digits of the given integer.		
		Write a program to obtain the gross salary by calculating dearness allowance and house rent.		
		Write a program to find the sum and average of ten different numbers using for loop,while loop and do while loop.		
		Write a program to search a number from 1 to 100 using break statement.		
		Write a program to print odd and even numbers from an array.		
		Write a program to find the highest and lowest number from 10 entered numbers		
		Write a program to find the G.C.D of the entered number.		
		Write a Java program to check whether a number is palindrome or not.		

	Write a Java program for following grading system. Note: Percentage $\geq$ 90% : Grade A Percentage $\geq$ 80% : Grade B Percentage $\geq$ 70% : Grade C Percentage $\geq$ 60% : Grade D Percentage $\geq$ 40% : Grade E Percentage		
	Write a program that will illustrate the concept of default constructor.		
	Create a class Person with properties(name and age) with following features: a. Default age of person should be 18. b. A person object can be initialized with name and age c. Method to display name and age of person.		
	d. Create a class Employee with(empNo ,salary and totalSalary)) with following features. e. a.Only parameterized constructor; f. b.totalSalary always represent total of all the salaries of all employees created. g. c.empNo should be auto incremented. h. d.display total employees and totalSalary using class method.		
	Write a program that will illustrate the concept of parameterized constructor and object parameterized constructor.		
	Write a program to illustrate the concept of this keyword.		
	Write a program to demonstrate static and instance data members and methods.		
	Write a Java program to implement the concept of importing classes from user defined package and creating packages		
	Write a program which illustrates the concept of inheritance.		
	Write a program to demonstrate the concept of method overloading and overriding.		
	Write a Java Program to demonstrate use of nested class.		
	Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it.		

		The class ThreeDObject has methods wholeSurfaceArea () and volume (). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three-dimensional objects. The dimensions of the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes. Write a program which illustrates the concept of super keyword at variable and method level. Write a program which will calculate the sum of two integer and two floats values using an abstract method sum() Write a program to demonstrate use of implementing and extends interfaces. Write program to demonstrate use of final keyword.		
3	Basic Pattern knowledge	Write a program to print the following pattern <pre> * * * * * * * * * * * * * * *</pre>		
		Write a program to print the following pattern <pre> 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 3 4 5 6</pre>		
		Write a program to print the following pattern <pre> 1 2 3 4 5 6 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1</pre>		
4	GUI	Write a program to print text into Text Area using		

Knowledge	JTextArea Create this frame Using JFrame  Write a program to display selected item from JCombo box in a label on click of Show button. Create a swing application to display a menu bar where each menu item should display a dialog box to show its working. Create this frame message will show according to radio button clicked.  Create this food ordering System.		

		Food Ordering System ☒ Pizza @ 100 ☒ Burger @ 30 ☒ Tea @ 10 Order Message i Pizza: 100 Burger: 30 Tea: 10 ----- Total: 140.0 OK		
--	--	--	--	--

INSTRUCTIONS TO STUDENTS

Before entering the lab the student should carry the following things
(MANDATORY)

1. Identity card issued by the college.
 2. Class notes
 3. Lab observation book
 4. Lab Manual
 5. Lab Record
- Student must sign in and sign out in the register provided when attending the lab session without fail.
 - Come to the laboratory in time. Students, who are late more than 15 min., will not be allowed to attend the lab.
 - Students need to maintain 100% attendance in lab if not a strict action will be taken.
 - All students must follow a Dress Code while in the laboratory.
 - All bags must be left at the indicated place.
 - Refer to the lab staff if you need any help in using the lab.
 - Respect the laboratory and its other users.
 - Workspace must be kept clean and tidy after experiment is completed.
 - Read the Manual carefully before coming to the laboratory and be sure about what you are supposed to do.
 - Do the experiments as per the instructions given in the manual.
 - Copy all the programs to observation which are taught in class before attending the lab session.
 - Students are not supposed to use pen drives without permission of lab- in

charge.

- Lab records need to be submitted on or before the date of submission.

Solutions

Assignment 1: Programs

1. Print "Hello JAVA"

java

CopyEdit

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

2. Print value from command line argument

java

CopyEdit

```
public class CommandLineInput {  
    public static void main(String[] args) {  
        for (String arg : args) {  
            System.out.println("Argument: " + arg);  
        }  
    }  
}
```

3. Sum of two numbers using command line

java

CopyEdit

```
public class SumCommandLine {  
    public static void main(String[] args) {  
        int a = Integer.parseInt(args[0]);  
        int b = Integer.parseInt(args[1]);  
        System.out.println("Sum: " + (a + b));  
    }  
}
```

4. Sum using Scanner class

java

CopyEdit

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

```
public class SumScanner {
```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter first number: ");
    int a = sc.nextInt();
    System.out.print("Enter second number: ");
    int b = sc.nextInt();
    System.out.println("Sum: " + (a + b));
}
}

```

5. Sum and average of 5 numbers

java

CopyEdit

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

```

public class SumAverage {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int sum = 0;
        System.out.println("Enter 5 numbers:");
        for (int i = 0; i < 5; i++) {
            sum += sc.nextInt();
        }
        double avg = sum / 5.0;
        System.out.println("Sum: " + sum);
        System.out.println("Average: " + avg);
    }
}

```

6. Swap using third variable

java

CopyEdit

```

public class SwapWithThird {
    public static void main(String[] args) {
        int a = 10, b = 20, temp;
        temp = a;
        a = b;
        b = temp;
    }
}

```

```
        System.out.println("a = " + a + ", b = " + b);
    }
}
```

6b. Swap without third variable

java

CopyEdit

```
public class SwapWithoutThird {
    public static void main(String[] args) {
        int a = 10, b = 20;
        a = a + b;
        b = a - b;
        a = a - b;
        System.out.println("a = " + a + ", b = " + b);
    }
}
```

7. Find greatest of 3 numbers

java

CopyEdit

```
import java.util.Scanner;

public class GreatestOfThree {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter 3 numbers:");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();
        int max = Math.max(a, Math.max(b, c));
        System.out.println("Greatest: " + max);
    }
}
```

8. Leap year check

java

CopyEdit

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

```

public class LeapYear {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int year = sc.nextInt();
        if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
            System.out.println(year + " is a Leap Year.");
        else
            System.out.println(year + " is not a Leap Year.");
    }
}

```

9. Factorial of a number

java

CopyEdit

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

```

public class Factorial {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        long fact = 1;
        for (int i = 1; i <= num; i++) {
            fact *= i;
        }
        System.out.println("Factorial: " + fact);
    }
}

```

10. Fibonacci series in range

java

CopyEdit

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

```

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

```

```

int a = 0, b = 1, c;
System.out.print("Fibonacci Series: " + a + " " + b);
for (int i = 2; i < range; i++) {
    c = a + b;
    System.out.print(" " + c);
    a = b;
    b = c;
}
}
}

```

11 Reverse of an integer

```

java
CopyEdit
import java.util.Scanner;

public class ReverseNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        int rev = 0;
        while (num != 0) {
            rev = rev * 10 + num % 10;
            num /= 10;
        }
        System.out.println("Reversed number: " + rev);
    }
}

```

12. Sum of digits of an integer

```

java
CopyEdit
import java.util.Scanner;

public class SumOfDigits {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        int sum = 0;
        while (num != 0) {
            sum += num % 10;
            num /= 10;
        }
    }
}

```

```
    }  
    System.out.println("Sum of digits: " + sum);  
}  
}
```

13. Gross salary = basic + DA + HRA

```
java  
CopyEdit  
import java.util.Scanner;  
  
public class GrossSalary {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter Basic Salary: ");  
        double basic = sc.nextDouble();  
        double da = 0.1 * basic;  
        double hra = 0.2 * basic;  
        double gross = basic + da + hra;  
        System.out.println("Gross Salary: " + gross);  
    }  
}
```

14. Sum and average using for, while, and do-while

```
java  
CopyEdit  
import java.util.Scanner;  
  
public class LoopsSumAverage {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int[] nums = new int[10];  
        System.out.println("Enter 10 numbers:");  
        for (int i = 0; i < 10; i++) nums[i] = sc.nextInt();  
  
        // Using for  
        int sum1 = 0;  
        for (int i = 0; i < 10; i++) sum1 += nums[i];  
        System.out.println("For loop => Sum: " + sum1 + ", Avg: " + sum1 / 10.0);  
  
        // Using while  
        int sum2 = 0, i = 0;  
        while (i < 10) sum2 += nums[i++];  
        System.out.println("While loop => Sum: " + sum2 + ", Avg: " + sum2 / 10.0);  
  
        // Using do-while  
        int sum3 = 0, j = 0;
```

```

do {
    sum3 += nums[j++];
} while (j < 10);
System.out.println("Do-While loop => Sum: " + sum3 + ", Avg: " + sum3 / 10.0);
}
}

```

15. Search number from 1 to 100 using break

```

java
CopyEdit
import java.util.Scanner;

public class BreakSearch {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int key = sc.nextInt();
        for (int i = 1; i <= 100; i++) {
            if (i == key) {
                System.out.println("Found at position: " + i);
                break;
            }
        }
    }
}

```

16. Print odd and even numbers from array

```

java
CopyEdit
import java.util.Scanner;

public class OddEvenArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[10];
        System.out.println("Enter 10 numbers:");
        for (int i = 0; i < 10; i++) arr[i] = sc.nextInt();

        System.out.print("Even numbers: ");
        for (int num : arr) if (num % 2 == 0) System.out.print(num + " ");
        System.out.print("\nOdd numbers: ");
        for (int num : arr) if (num % 2 != 0) System.out.print(num + " ");
    }
}

```

17. Highest and lowest from 10 numbers

```

java

```

CopyEdit

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

```
public class MaxMin {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;

        System.out.println("Enter 10 numbers:");
        for (int i = 0; i < 10; i++) {
            int num = sc.nextInt();
            if (num > max) max = num;
            if (num < min) min = num;
        }

        System.out.println("Max: " + max);
        System.out.println("Min: " + min);
    }
}
```

18. GCD of two numbers

java

CopyEdit

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

```
public class GCD {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int b = sc.nextInt();
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        System.out.println("GCD: " + a);
    }
}
```

19. Check if number is palindrome

java

CopyEdit

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

```
public class PalindromeCheck {
```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int num = sc.nextInt();
    int original = num, reverse = 0;
    while (num != 0) {
        reverse = reverse * 10 + num % 10;
        num /= 10;
    }
    if (original == reverse)
        System.out.println("Palindrome");
    else
        System.out.println("Not Palindrome");
}
}

```

20. Grading system

```

java
CopyEdit
import java.util.Scanner;

public class GradeSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double percent = sc.nextDouble();

        if (percent >= 90)
            System.out.println("Grade A");
        else if (percent >= 80)
            System.out.println("Grade B");
        else if (percent >= 70)
            System.out.println("Grade C");
        else if (percent >= 60)
            System.out.println("Grade D");
        else if (percent >= 40)
            System.out.println("Grade E");
        else
            System.out.println("Fail");
    }
}

```

21. Parameterized constructor and object parameter constructor

```

java
CopyEdit
class Student {
    String name;
    int age;
}

```

```

Student(String n, int a) {
    name = n;
    age = a;
}

Student(Student s) {
    name = s.name;
    age = s.age;
}

void display() {
    System.out.println("Name: " + name + ", Age: " + age);
}

public static void main(String[] args) {
    Student s1 = new Student("Ankit", 20);
    Student s2 = new Student(s1);
    s1.display();
    s2.display();
}
}

```

22. this keyword usage

```

java
CopyEdit
class ThisExample {
    int x;

    ThisExample(int x) {
        this.x = x; // refers to current object
    }

    void display() {
        System.out.println("x = " + this.x);
    }

    public static void main(String[] args) {
        ThisExample obj = new ThisExample(10);
        obj.display();
    }
}

```

23. Static and instance members

```

java
CopyEdit
class StaticDemo {

```

```

int instanceVar = 10;
static int staticVar = 20;

void show() {
    System.out.println("Instance var: " + instanceVar);
    System.out.println("Static var: " + staticVar);
}

static void staticMethod() {
    System.out.println("Static var from static method: " + staticVar);
}

public static void main(String[] args) {
    StaticDemo obj = new StaticDemo();
    obj.show();
    StaticDemo.staticMethod();
}
}

```

24. Importing and using user-defined package

Package: mypack/Message.java

```

java
CopyEdit
package mypack;

public class Message {
    public void show() {
        System.out.println("Hello from user-defined package!");
    }
}

```

Main Class: TestPackage.java

```

java
CopyEdit
import mypack.Message;

public class TestPackage {
    public static void main(String[] args) {
        Message msg = new Message();
        msg.show();
    }
}

```

25. Inheritance example

```

java
CopyEdit
class Animal {

```

```

    void sound() {
        System.out.println("Animal makes sound");
    }
}

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

public class InheritanceDemo {
    public static void main(String[] args) {
        Dog d = new Dog();
        d.sound();
    }
}

```

26. Method Overloading and Overriding

```

java
CopyEdit
class Overload {
    void display(int a) {
        System.out.println("int: " + a);
    }

    void display(String s) {
        System.out.println("String: " + s);
    }
}

class Override extends Overload {
    @Override
    void display(String s) {
        System.out.println("Overridden String: " + s);
    }
}

public class OverloadOverride {
    public static void main(String[] args) {
        Override obj = new Override();
        obj.display(5);
        obj.display("Java");
    }
}

```

27. Nested class example

```
java
CopyEdit
public class Outer {
    int x = 10;

    class Inner {
        void show() {
            System.out.println("Inner class, x = " + x);
        }
    }

    public static void main(String[] args) {
        Outer o = new Outer();
        Outer.Inner in = o.new Inner();
        in.show();
    }
}
```

28. ThreeDObject → Box, Cube, Cylinder, Cone

```
java
CopyEdit
abstract class ThreeDObject {
    abstract double volume();
    abstract double surfaceArea();
}

class Box extends ThreeDObject {
    double l, b, h;
    Box(double l, double b, double h) {
        this.l = l;
        this.b = b;
        this.h = h;
    }
    double volume() { return l * b * h; }
    double surfaceArea() { return 2 * (l*b + b*h + h*l); }
}

class Cube extends ThreeDObject {
    double side;
    Cube(double s) { side = s; }
    double volume() { return side * side * side; }
    double surfaceArea() { return 6 * side * side; }
}

class Cylinder extends ThreeDObject {
```

```

double r, h;
Cylinder(double r, double h) {
    this.r = r;
    this.h = h;
}
double volume() { return Math.PI * r * r * h; }
double surfaceArea() { return 2 * Math.PI * r * (r + h); }
}

class Cone extends ThreeDObject {
    double r, h;
    Cone(double r, double h) {
        this.r = r;
        this.h = h;
    }
    double volume() { return (1.0 / 3) * Math.PI * r * r * h; }
    double surfaceArea() {
        double l = Math.sqrt(r * r + h * h);
        return Math.PI * r * (r + l);
    }
}

public class Test3D {
    public static void main(String[] args) {
        ThreeDObject[] shapes = {
            new Box(2, 3, 4),
            new Cube(5),
            new Cylinder(2, 6),
            new Cone(2, 5)
        };

        for (ThreeDObject obj : shapes) {
            System.out.println("Volume: " + obj.volume());
            System.out.println("Surface Area: " + obj.surfaceArea());
            System.out.println("---");
        }
    }
}

```

29. super keyword example

```

java
CopyEdit
class Parent {
    int num = 100;
    void display() {
        System.out.println("Parent method");
    }
}

```

```

    }
}

class Child extends Parent {
    int num = 200;
    void display() {
        super.display();
        System.out.println("Child method");
        System.out.println("Parent num: " + super.num);
        System.out.println("Child num: " + num);
    }

    public static void main(String[] args) {
        new Child().display();
    }
}

```

30. Abstract class with sum() method

```

java
CopyEdit
abstract class Adder {
    abstract int sum(int a, int b);
    abstract float sum(float a, float b);
}

class Calculator extends Adder {
    int sum(int a, int b) {
        return a + b;
    }

    float sum(float a, float b) {
        return a + b;
    }

    public static void main(String[] args) {
        Calculator c = new Calculator();
        System.out.println("Int Sum: " + c.sum(10, 20));
        System.out.println("Float Sum: " + c.sum(1.5f, 2.3f));
    }
}

```

31. Implements and extends interface

```

java
CopyEdit
interface A {
    void show();
}

```

```
}

class B {
    void display() {
        System.out.println("Display from class B");
    }
}

class C extends B implements A {
    public void show() {
        System.out.println("Show from interface A");
    }

    public static void main(String[] args) {
        C obj = new C();
        obj.display();
        obj.show();
    }
}
```

32. Final keyword usage

```
java
CopyEdit
final class Constant {
    final int VALUE = 100;

    void show() {
        System.out.println("Final VALUE = " + VALUE);
    }
}

public class FinalDemo {
    public static void main(String[] args) {
        Constant c = new Constant();
        c.show();
    }
}
```

```
*
* *
* * *
* * * *
* * * * *
```

java

CopyEdit

```
public class Pattern1 {
    public static void main(String[] args) {
        for(int i = 1; i <= 5; i++) {
            for(int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

12. Pattern:

CopyEdit

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

java

CopyEdit

```
public class Pattern2 {
    public static void main(String[] args) {
        for(int i = 1; i <= 6; i++) {
            for(int j = 1; j <= i; j++) {
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }
}
```

13. Pattern:

CopyEdit

```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
```

```
1
java
CopyEdit
public class Pattern3 {
    public static void main(String[] args) {
        for(int i = 6; i >= 1; i--) {
            for(int j = 1; j <= i; j++) {
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }
}
```

Print text into TextArea using JTextArea

```
java
CopyEdit
import javax.swing.*;

public class TextAreaExample {
    public static void main(String[] args) {
        JFrame f = new JFrame("TextArea Example");
        JTextArea area = new JTextArea("Welcome to Java Swing!");
        area.setBounds(10, 30, 300, 100);
        f.add(area);
        f.setSize(400, 200);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

2. Create Frame using JFrame

```
java
CopyEdit
import javax.swing.*;

public class MyFrame {
    public static void main(String[] args) {
        JFrame frame = new JFrame("My JFrame Example");
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

3. Display selected item from JComboBox in a JLabel

```
java
CopyEdit
import javax.swing.*;
import java.awt.event.*;

public class ComboBoxExample {
    public static void main(String[] args) {
        JFrame f = new JFrame("ComboBox Example");
        String[] items = {"Java", "Python", "C++"};
        JComboBox<String> cb = new JComboBox<>(items);
        cb.setBounds(50, 50, 100, 30);

        JLabel label = new JLabel();
        label.setBounds(50, 100, 200, 30);

        JButton b = new JButton("Show");
        b.setBounds(160, 50, 80, 30);

        b.addActionListener(e -> {
            String selected = (String) cb.getSelectedItem();
            label.setText("Selected: " + selected);
        });

        f.add(cb);
        f.add(b);
        f.add(label);
        f.setSize(300, 200);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

4. MenuBar with dialogs for each menu item

```
java
CopyEdit
import javax.swing.*;
import java.awt.event.*;

public class MenuDialog {
    public static void main(String[] args) {
        JFrame f = new JFrame("Menu Example");

        JMenuBar mb = new JMenuBar();
        JMenu menu = new JMenu("Options");
        JMenuItem about = new JMenuItem("About");
```

```

JMenuItem exit = new JMenuItem("Exit");

about.addActionListener(e -> JOptionPane.showMessageDialog(f, "This is a Java Swing menu."));
exit.addActionListener(e -> System.exit(0));

menu.add(about);
menu.add(exit);
mb.add(menu);

f.setJMenuBar(mb);
f.setSize(300, 200);
f.setLayout(null);
f.setVisible(true);
}
}

```

5. Radio button gender selection with message

```

java
CopyEdit
import javax.swing.*;
import java.awt.event.*;

public class RadioButtonExample {
    public static void main(String[] args) {
        JFrame f = new JFrame("Radio Button Example");
        JRadioButton male = new JRadioButton("Male");
        JRadioButton female = new JRadioButton("Female");
        male.setBounds(50, 50, 100, 30);
        female.setBounds(50, 80, 100, 30);

        ButtonGroup bg = new ButtonGroup();
        bg.add(male);
        bg.add(female);

        JButton b = new JButton("Show");
        b.setBounds(50, 120, 100, 30);
        JLabel l = new JLabel();
        l.setBounds(50, 160, 200, 30);

        b.addActionListener(e -> {
            if (male.isSelected()) l.setText("Selected: Male");
            else if (female.isSelected()) l.setText("Selected: Female");
            else l.setText("No selection");
        });

        f.add(male);
    }
}

```

```

        f.add(female);
        f.add(b);
        f.add(l);
        f.setSize(300, 250);
        f.setLayout(null);
        f.setVisible(true);
    }
}

```

6. Food Ordering System (Pizza ₹100, Burger ₹30, Tea ₹10)

```

java
CopyEdit
import javax.swing.*;
import java.awt.event.*;

public class FoodOrderingSystem {
    public static void main(String[] args) {
        JFrame f = new JFrame("Food Ordering System");

        JCheckBox pizza = new JCheckBox("Pizza @ ₹100");
        JCheckBox burger = new JCheckBox("Burger @ ₹30");
        JCheckBox tea = new JCheckBox("Tea @ ₹10");

        pizza.setBounds(50, 50, 150, 30);
        burger.setBounds(50, 80, 150, 30);
        tea.setBounds(50, 110, 150, 30);

        JButton b = new JButton("Order");
        b.setBounds(50, 150, 100, 30);

        JLabel result = new JLabel("Total: ₹0");
        result.setBounds(50, 190, 200, 30);

        b.addActionListener(e -> {
            int amount = 0;
            if (pizza.isSelected()) amount += 100;
            if (burger.isSelected()) amount += 30;
            if (tea.isSelected()) amount += 10;
            result.setText("Total: ₹" + amount);
        });

        f.add(pizza);
        f.add(burger);
        f.add(tea);
        f.add(b);
        f.add(result);
        f.setSize(300, 300);
    }
}

```

```
f.setLayout(null);  
f.setVisible(true);  
}  
}
```