

1. Hello World Program

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

Hello, World!

2. Simple Calculator

```
import java.util.Scanner;  
  
public class Calculator {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter first number: ");  
        double a = sc.nextDouble();  
        System.out.print("Enter second number: ");  
        double b = sc.nextDouble();  
        System.out.print("Choose operation (+, -, *, /): ");  
        char op = sc.next().charAt(0);  
  
        double result;  
        switch (op) {  
            case '+': result = a + b; break;  
            case '-': result = a - b; break;  
            case '*': result = a * b; break;  
            case '/': result = b != 0 ? a / b : 0; break;  
            default: result = 0;  
        }  
        System.out.println("Result: " + result);  
    }  
}
```

Enter first number: 5
Enter second number: 2
Choose operation (+, -, *, /): *
Result: 10.0

3. Even or Odd Checker

```
import java.util.Scanner;

public class EvenOdd {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int num = sc.nextInt();
        System.out.println(num % 2 == 0 ? "Even" : "Odd");
    }
}
```

Enter an integer: 7
Odd

4. Leap Year Checker

```
import java.util.Scanner;

public class LeapYear {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter year: ");
        int year = sc.nextInt();
        boolean isLeap = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
        System.out.println(isLeap ? "Leap Year" : "Not a Leap Year");
    }
}
```

Enter year: 2024
Leap Year

5. Multiplication Table

```
import java.util.Scanner;

public class Table {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        for (int i = 1; i <= 10; i++)
```

```

        System.out.println(num + " x " + i + " = " + (num * i));
    }
}

```

Enter a number: 3

3 x 1 = 3

...

3 x 10 = 30

6. Data Type Demonstration

```

public class DataTypes {
    public static void main(String[] args) {
        int a = 100;
        float b = 12.5f;
        double c = 123.456;
        char d = 'A';
        boolean e = true;

        System.out.println("int: " + a);
        System.out.println("float: " + b);
        System.out.println("double: " + c);
        System.out.println("char: " + d);
        System.out.println("boolean: " + e);
    }
}

```

int: 100

float: 12.5

double: 123.456

char: A

boolean: true

7. Type Casting Example

```

public class TypeCasting {
    public static void main(String[] args) {
        double d = 9.7;
        int i = (int) d;
        int x = 20;
        double y = x;

        System.out.println("Double to int: " + i);
        System.out.println("Int to double: " + y);
    }
}

```

```
}  
}
```

Double to int: 9
Int to double: 20.0

8. Operator Precedence

```
public class Precedence {  
    public static void main(String[] args) {  
        int result = 10 + 5 * 2;  
        System.out.println("Result: " + result); // Multiplication before addition  
    }  
}
```

Result: 20

9. Grade Calculator

```
import java.util.Scanner;  
  
public class GradeCalc {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter marks out of 100: ");  
        int marks = sc.nextInt();  
        char grade;  
  
        if (marks >= 90) grade = 'A';  
        else if (marks >= 80) grade = 'B';  
        else if (marks >= 70) grade = 'C';  
        else if (marks >= 60) grade = 'D';  
        else grade = 'F';  
  
        System.out.println("Grade: " + grade);  
    }  
}
```

Enter marks out of 100: 82
Grade: B

10. Number Guessing Game

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

```

public class GuessGame {
    public static void main(String[] args) {
        int number = (int) (Math.random() * 100 + 1);
        Scanner sc = new Scanner(System.in);
        int guess = 0;

        while (guess != number) {
            System.out.print("Guess a number (1-100): ");
            guess = sc.nextInt();
            if (guess < number) System.out.println("Too low!");
            else if (guess > number) System.out.println("Too high!");
        }
        System.out.println("Correct! The number was " + number);
    }
}

```

```

Guess a number (1-100): 50
                        Too high!
Guess a number (1-100): 25
                        Too low!
Guess a number (1-100): 37
Correct! The number was 37

```

11. Factorial Calculator

```

import java.util.Scanner;

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

```

```

Enter a number: 5
Factorial of 5 = 120

```

12. Method Overloading

```
public class Overload {  
    public int add(int a, int b) { return a + b; }  
    public double add(double a, double b) { return a + b; }  
    public int add(int a, int b, int c) { return a + b + c; }  
  
    public static void main(String[] args) {  
        Overload obj = new Overload();  
        System.out.println("add(int, int): " + obj.add(5, 3));  
        System.out.println("add(double, double): " + obj.add(2.5, 4.5));  
        System.out.println("add(int, int, int): " + obj.add(1, 2, 3));  
    }  
}
```

add(int, int): 8
add(double, double): 7.0
add(int, int, int): 6

13. Recursive Fibonacci

```
import java.util.Scanner;  
  
public class Fibonacci {  
    static int fibonacci(int n) {  
        if (n <= 1) return n;  
        return fibonacci(n - 1) + fibonacci(n - 2);  
    }  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter n: ");  
        int n = sc.nextInt();  
        System.out.println(n + "th Fibonacci number is: " + fibonacci(n));  
    }  
}
```

Enter n: 6
6th Fibonacci number is: 8

14. Array Sum and Average

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

```

public class ArrayStats {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        int sum = 0;

        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
            sum += arr[i];
        }
        double avg = (double) sum / n;
        System.out.println("Sum: " + sum + ", Average: " + avg);
    }
}

```

Enter number of elements: 3

Enter elements:

10

20

30

Sum: 60, Average: 20.0

15. String Reversal

```

import java.util.Scanner;

public class ReverseString {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter string: ");
        String str = sc.nextLine();
        String reversed = new StringBuilder(str).reverse().toString();
        System.out.println("Reversed: " + reversed);
    }
}

```

Enter string: Hello

Reversed: olleH

16. Palindrome Checker

```
import java.util.Scanner;

public class Palindrome {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter string: ");
        String original = sc.nextLine().replaceAll("[^a-zA-Z0-9]", "").toLowerCase();
        String reversed = new StringBuilder(original).reverse().toString();
        System.out.println(original.equals(reversed) ? "Palindrome" : "Not Palindrome");
    }
}
```

Enter string: Madam
Palindrome

17. Class and Object Creation

```
class Car {
    String make, model;
    int year;

    void displayDetails() {
        System.out.println(year + " " + make + " " + model);
    }
}

public class CarTest {
    public static void main(String[] args) {
        Car c1 = new Car();
        c1.make = "Toyota";
        c1.model = "Corolla";
        c1.year = 2022;
        c1.displayDetails();
    }
}
```

2022 Toyota Corolla

18. Inheritance Example

```
class Animal {
```



```

    void makeSound() {
        System.out.println("Some sound");
    }
}

```

```

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

```

```

public class TestAnimal {
    public static void main(String[] args) {
        Animal a = new Animal();
        Dog d = new Dog();
        a.makeSound();
        d.makeSound();
    }
}

```

Some sound
Bark

19. Interface Implementation

```

interface Playable {
    void play();
}

```

```

class Guitar implements Playable {
    public void play() {
        System.out.println("Playing Guitar");
    }
}

```

```

class Piano implements Playable {
    public void play() {
        System.out.println("Playing Piano");
    }
}

```

```

public class InstrumentTest {

```

```

public static void main(String[] args) {
    Playable g = new Guitar();
    Playable p = new Piano();
    g.play();
    p.play();
}
}

```

Playing Guitar
Playing Piano

20. Try-Catch Example

```

import java.util.Scanner;

public class DivisionTryCatch {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter first number: ");
            int a = sc.nextInt();
            System.out.print("Enter second number: ");
            int b = sc.nextInt();
            System.out.println("Result: " + (a / b));
        } catch (ArithmeticException e) {
            System.out.println("Cannot divide by zero.");
        }
    }
}

```

Enter first number: 10
Enter second number: 0
Cannot divide by zero.

21. Custom Exception

```

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

```

```

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

```

```

        if (age < 18) {
            throw new InvalidAgeException("Age must be 18 or above.");
        }
        System.out.println("Access granted.");
    } catch (InvalidAgeException e) {
        System.out.println("Exception: " + e.getMessage());
    }
}
}

```

Exception: Age must be 18 or above.

22. File Writing

```

import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class WriteToFile {
    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter text to write: ");
            String text = sc.nextLine();

            FileWriter fw = new FileWriter("output.txt");
            fw.write(text);
            fw.close();
            System.out.println("Data written to output.txt");
        } catch (IOException e) {
            System.out.println("Error writing file: " + e.getMessage());
        }
    }
}

```

Enter text to write: Hello Java
Data written to output.txt

23. File Reading

```

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class ReadFromFile {

```

```

public static void main(String[] args) {
    try {
        BufferedReader br = new BufferedReader(new FileReader("output.txt"));
        String line;
        while ((line = br.readLine()) != null)
            System.out.println(line);
        br.close();
    } catch (IOException e) {
        System.out.println("Error reading file.");
    }
}
}

```

Hello Java

24. ArrayList Example

```

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

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

        System.out.println("Enter 3 student names:");
        for (int i = 0; i < 3; i++) {
            names.add(sc.nextLine());
        }

        System.out.println("Student List:");
        for (String name : names) {
            System.out.println(name);
        }
    }
}

```

Enter 3 student names:

Alice

Bob

Charlie

Student List:

Alice

25. HashMap Example

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

public class StudentMap {
    public static void main(String[] args) {
        HashMap<Integer, String> studentMap = new HashMap<>();
        Scanner sc = new Scanner(System.in);

        studentMap.put(1, "Alice");
        studentMap.put(2, "Bob");

        System.out.print("Enter student ID to search: ");
        int id = sc.nextInt();
        System.out.println("Name: " + studentMap.getOrDefault(id, "Not Found"));
    }
}
```

Enter student ID to search: 2
Name: Bob

26. Thread Creation

```
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++)
            System.out.println(Thread.currentThread().getName() + " - Count: " + i);
    }
}

public class ThreadTest {
    public static void main(String[] args) {
        MyThread t1 = new MyThread();
        MyThread t2 = new MyThread();
        t1.start();
        t2.start();
    }
}
```

Thread-0 - Count: 1
Thread-1 - Count: 1

27. Lambda Expressions

```
import java.util.*;

public class LambdaSort {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Charlie", "Alice", "Bob");
        Collections.sort(names, (a, b) -> a.compareTo(b));
        System.out.println("Sorted: " + names);
    }
}
```

Sorted: [Alice, Bob, Charlie]

28. Stream API

```
import java.util.*;
import java.util.stream.*;

public class StreamEven {
    public static void main(String[] args) {
        List<Integer> nums = Arrays.asList(1, 2, 3, 4, 5, 6);
        List<Integer> evens = nums.stream().filter(n -> n % 2 ==
0).collect(Collectors.toList());
        System.out.println("Even numbers: " + evens);
    }
}
```

Even numbers: [2, 4, 6]

29. Records (Java 16+)

```
record Person(String name, int age) {}

import java.util.*;
import java.util.stream.*;

public class RecordTest {
    public static void main(String[] args) {
        List<Person> people = List.of(
            new Person("Alice", 25),
            new Person("Bob", 30),
            new Person("Charlie", 17)
        );
    }
}
```

```

    );
    people.stream().filter(p -> p.age() >= 18).forEach(System.out::println);
}
}

```

Person[name=Alice, age=25]

Person[name=Bob, age=30]

30. Pattern Matching for switch (Java 21)

```

public class PatternSwitch {
    public static void main(Object obj) {
        switch (obj) {
            case Integer i -> System.out.println("Integer: " + i);
            case String s -> System.out.println("String: " + s);
            case Double d -> System.out.println("Double: " + d);
            default -> System.out.println("Unknown type");
        }
    }

    public static void main(String[] args) {
        main("Hello");
        main(100);
        main(12.34);
    }
}

```

String: Hello

Integer: 100

Double: 12.34

31. Basic JDBC Connection

```

import java.sql.*;

public class JDBCSelect {
    public static void main(String[] args) {
        try {
            Connection con = DriverManager.getConnection("jdbc:sqlite:students.db");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM students");

            while (rs.next()) {
                System.out.println(rs.getInt("id") + " - " + rs.getString("name"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
    con.close();
} catch (SQLException e) {
    System.out.println("Connection failed: " + e.getMessage());
}
}
}

```

1 - Alice

2 – Bob

32. Insert and Update Operations in JDBC

```

import java.sql.*;

public class StudentDAO {
    public static void insertStudent(String name) throws SQLException {
        Connection con = DriverManager.getConnection("jdbc:sqlite:students.db");
        String sql = "INSERT INTO students(name) VALUES(?)";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, name);
        ps.executeUpdate();
        con.close();
    }

    public static void updateStudent(int id, String name) throws SQLException {
        Connection con = DriverManager.getConnection("jdbc:sqlite:students.db");
        String sql = "UPDATE students SET name=? WHERE id=?";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, name);
        ps.setInt(2, id);
        ps.executeUpdate();
        con.close();
    }
}

```

Record inserted/updated (via method call, no direct console output)

33. Transaction Handling in JDBC

```

import java.sql.*;

public class TransferMoney {
    public static void transfer(int fromId, int told, double amount) {

```



```

try (Connection con = DriverManager.getConnection("jdbc:sqlite:bank.db")) {
    con.setAutoCommit(false);

    PreparedStatement debit = con.prepareStatement("UPDATE accounts SET
balance = balance - ? WHERE id = ?");
    debit.setDouble(1, amount);
    debit.setInt(2, fromId);
    debit.executeUpdate();

    PreparedStatement credit = con.prepareStatement("UPDATE accounts SET
balance = balance + ? WHERE id = ?");
    credit.setDouble(1, amount);
    credit.setInt(2, toId);
    credit.executeUpdate();

    con.commit();
    System.out.println("Transfer successful.");
} catch (SQLException e) {
    System.out.println("Transfer failed: " + e.getMessage());
}
}
}

```

Transfer successful.

34. Create and Use Java Modules

Utils.java

```

package com.utils;

public class Utils {
    public static String greet(String name) {
        return "Hello, " + name;
    }
}

```

module-info.java

```

module com.utils {
    exports com.utils;
}

```

Main.java

```

package com.greetings;

import com.utils.Utils;

```

```

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

```

module-info.java

```

module com.greetings {
    requires com.utils;
}

```

Hello, World

35. TCP Client-Server Chat

Server.java

```

import java.io.*;
import java.net.*;

public class Server {
    public static void main(String[] args) throws IOException {
        ServerSocket ss = new ServerSocket(1234);
        Socket s = ss.accept();
        BufferedReader br = new BufferedReader(new
InputStreamReader(s.getInputStream()));
        PrintWriter pw = new PrintWriter(s.getOutputStream(), true);

        pw.println("Hello from server!");
        System.out.println("Client says: " + br.readLine());

        ss.close();
    }
}

```

Client.java

```

import java.io.*;
import java.net.*;

public class Client {
    public static void main(String[] args) throws IOException {
        Socket s = new Socket("localhost", 1234);
        BufferedReader br = new BufferedReader(new
InputStreamReader(s.getInputStream()));
        PrintWriter pw = new PrintWriter(s.getOutputStream(), true);
    }
}

```

```

        System.out.println("Server says: " + br.readLine());
        pw.println("Hello from client!");
        s.close();
    }
}

```

Server: Client says: Hello from client!

Client: Server says: Hello from server!

36. HTTP Client API (Java 11+)

```

import java.net.http.*;
import java.net.URI;

public class HttpExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(new URI("https://api.github.com"))
            .build();

        HttpResponse<String> response = client.send(request,
            HttpResponse.BodyHandlers.ofString());

        System.out.println("Status: " + response.statusCode());
        System.out.println("Body: " + response.body().substring(0, 100) + "...");
    }
}

```

Status: 200

Body: { "current_user_url": "https://api.github.com/user", ... }

37. Using javap to Inspect Bytecode

```

public class Test {
    public void hello() {
        System.out.println("Hello Bytecode");
    }
}

```

javac Test.java

javap -c Test

public void hello();

Code:

```

0: getstatic    #2          // Field java/lang/System.out:Ljava/io/PrintStream;
3: ldc          #3          // String Hello Bytecode
5: invokevirtual #4          // Method java/io/PrintStream.println
8: return

```

38. Decompile a Class File

```

public class Test {
    public void hello() {
        System.out.println("Hello Bytecode");
    }
}

```

39. Reflection in Java

```

import java.lang.reflect.*;

public class ReflectionDemo {
    public static void main(String[] args) throws Exception {
        Class<?> cls = Class.forName("java.lang.String");
        Method[] methods = cls.getDeclaredMethods();
        for (Method m : methods) {
            System.out.println(m.getName());
        }
    }
}

```

```

substring
length
charAt
...

```

40. Virtual Threads (Java 21)

```

public class VirtualThreadTest {
    public static void main(String[] args) {
        for (int i = 0; i < 100_000; i++) {
            Thread.startVirtualThread(() -> {
                System.out.println("Running: " + Thread.currentThread());
            });
        }
    }
}

```

```

Running: VirtualThread[#...]
...

```

41. ExecutorService and Callable

```
import java.util.concurrent.*;

public class CallableExample {
    public static void main(String[] args) throws Exception {
        ExecutorService es = Executors.newFixedThreadPool(2);

        Callable<String> task1 = () -> "Result from Task 1";
        Callable<String> task2 = () -> "Result from Task 2";

        Future<String> f1 = es.submit(task1);
        Future<String> f2 = es.submit(task2);

        System.out.println(f1.get());
        System.out.println(f2.get());

        es.shutdown();
    }
}
```

Result from Task 1

Result from Task 2