**1. Hello World Program**

**Class:** HelloWorld

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**2. Simple Calculator**

**Class:** SimpleCalculator  
Use Scanner to get user input, switch-case for operations.

**3. Even or Odd Checker**

**Class:** EvenOddChecker

int number = scanner.nextInt();

System.out.println(number % 2 == 0 ? "Even" : "Odd");

**4. Leap Year Checker**

**Class:** LeapYearChecker  
Check: (year % 4 == 0 && year % 100 != 0) || year % 400 == 0

**5. Multiplication Table**

**Class:** MultiplicationTable

for (int i = 1; i <= 10; i++) System.out.println(num + " x " + i + " = " + (num \* i));

**6. Data Type Demonstration**

**Class:** DataTypeDemo  
Declare and print: int, float, double, char, boolean

**7. Type Casting Example**

**Class:** TypeCastingExample

double d = 9.8;

int i = (int) d;

System.out.println(i);

**8. Operator Precedence**

**Class:** OperatorPrecedenceDemo

int result = 10 + 5 \* 2; // Output: 20

**9. Grade Calculator**

**Class:** GradeCalculator  
Use if-else to assign grades from marks.

**10. Number Guessing Game**

**Class:** GuessingGame  
Use Random and while loop until guessed.

**11. Factorial Calculator**

**Class:** FactorialCalculator

int fact = 1;

for (int i = 1; i <= num; i++) fact \*= i;

**12. Method Overloading**

**Class:** MethodOverloadDemo

int add(int a, int b) { return a + b; }

double add(double a, double b) { return a + b; }

int add(int a, int b, int c) { return a + b + c; }

**13. Recursive Fibonacci**

**Class:** RecursiveFibonacci

int fib(int n) {

if (n <= 1) return n;

return fib(n - 1) + fib(n - 2);

}

**14. Array Sum and Average**

**Class:** ArrayStats

int sum = Arrays.stream(arr).sum();

double avg = (double) sum / arr.length;

**15. String Reversal**

**Class:** StringReverser

String reversed = new StringBuilder(input).reverse().toString();

**16. Palindrome Checker**

**Class:** PalindromeChecker

String clean = str.replaceAll("[^a-zA-Z0-9]", "").toLowerCase();

boolean isPalindrome = clean.equals(new StringBuilder(clean).reverse().toString());

**17. Class and Object Creation**

**Class:** Car

class Car {

String make, model;

int year;

void displayDetails() { ... }

}

**18. Inheritance Example**

**Class:** Animal, Dog

class Animal { void makeSound() { ... } }

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

**19. Interface Implementation**

**Interface:** Playable, Classes: Guitar, Piano

**20. Try-Catch Example**

**Class:** DivisionHandler

try {

int result = a / b;

} catch (ArithmeticException e) {

System.out.println("Cannot divide by zero.");

}

**21. Custom Exception**

**Class:** InvalidAgeException  
Throw if age < 18.

**22. File Writing**

**Class:** FileWriterDemo

Files.writeString(Paths.get("output.txt"), input);

**23. File Reading**

**Class:** FileReaderDemo

Files.lines(Paths.get("output.txt")).forEach(System.out::println);

**24. ArrayList Example**

**Class:** StudentListDemo

List<String> names = new ArrayList<>();

names.add("Alice");

**25. HashMap Example**

**Class:** StudentMapDemo

Map<Integer, String> studentMap = new HashMap<>();

**26. Thread Creation**

**Class:** MyThread extends Thread or implements Runnable

**27. Lambda Expressions**

**Class:** LambdaSortDemo

Collections.sort(list, (a, b) -> a.compareTo(b));

**28. Stream API**

**Class:** EvenFilter

list.stream().filter(n -> n % 2 == 0).forEach(System.out::println);

**29. Records**

**Class:** record Person(String name, int age)  
Filter using Stream: list.stream().filter(p -> p.age() > 20)...

**30. Pattern Matching for switch (Java 21+)**

**Class:** PatternSwitch

switch (obj) {

case String s -> System.out.println("It's a String: " + s);

case Integer i -> System.out.println("Integer: " + i);

}

**31. Basic JDBC Connection**

**Class:** JDBCConnect  
Use Connection, DriverManager, ResultSet

**32. Insert and Update in JDBC**

**Class:** StudentDAO

PreparedStatement ps = conn.prepareStatement("INSERT INTO students ...");

**33. Transaction Handling in JDBC**

**Class:** BankTransfer

conn.setAutoCommit(false);

// Do debit/credit

conn.commit(); // or conn.rollback()

**34. Java Modules**

* module-info.java in both com.greetings and com.utils
* Export utility class and use in other module

**35. TCP Client-Server Chat**

Use ServerSocket, Socket, InputStreamReader, PrintWriter

**36. HTTP Client API**

**Class:** HttpClientDemo

HttpClient client = HttpClient.newHttpClient();

HttpRequest request = HttpRequest.newBuilder().uri(URI.create("https://api.github.com")).build();

**37. javap Bytecode Inspection**

javac HelloWorld.java

javap -c HelloWorld

**38. Decompile a Class File**

Use JD-GUI or CFR to open .class file and view source.

**39. Reflection in Java**

**Class:** ReflectionDemo

Class<?> cls = Class.forName("MyClass");

Method[] methods = cls.getDeclaredMethods();

**40. Virtual Threads (Java 21)**

for (int i = 0; i < 100000; i++) {

Thread.startVirtualThread(() -> System.out.println("Running..."));

}

**41. ExecutorService and Callable**

ExecutorService exec = Executors.newFixedThreadPool(3);

List<Callable<Integer>> tasks = ...

List<Future<Integer>> results = exec.invokeAll(tasks);