JAVA MASTERY IN 4 DAYS: DETAILED TEACHING AGENDA

sDay	Topic	Sub-Topics	Time	Code Examples / Homework
Day 1	Introduction to Java	Overview of Java, Features and Benefits, Setting up the Java Development Environment	20 minutes	

	Basic Syntax	Writing a Simple Java Program, Explanation of Main Method, Compiling and Running a Java Program	30 minutes	java
				public class HelloWorld {
				 public static void main(String[] args) {
				 System.out.println("Hello, World!");
				 }
				}

	Data Types and Variables	Primitive Data Types, Declaring and Initializing Variables, Type Casting	30 minutes	java
				int age = 25;
				double salary = 50000.75;
				char grade = 'A';
				boolean isJavaFun = true;

	Control Statements	If-Else Statements, Switch-Case Statements, Loops (for, while, do-while), Break and Continue Statements	40 minutes	java
				int number = 10;
				if (number > 0) {
				 System.out.println("Positive number");
				} else {
				 System.out.println("Non-positive number");
				}

			Homework	Simple exercises to practice basic syntax and control statements
Day 2	Introduction to OOP	Concepts of Objects and Classes, Benefits of OOP	20 minutes	

	Classes and Objects	Defining a Class, Creating Objects, Constructors	30 minutes	java
				public class Person {
				 String name;
				 int age;
				}
				Person person1 = new Person();
				person1.name = "John";
				person1.age = 30;

	Methods	Defining and Calling Methods, Method Overloading	30 minutes	java
				public class Calculator {
				 public int add(int a, int b) {
				 return a + b;
				 }
				}
				Calculator calc = new Calculator();
				int sum = calc.add(5, 10);
				System.out.println(sum);

	Encapsulation	Access Modifiers, Getters and Setters	40 minutes	java
				public class Person {
				 private String name;
				 private int age;
				 public String getName() {
				 return name;
				 }
				 public void setName(String name) {
				 this.name = name;
				 }
				 public int getAge() {
				 return age;
				 }
				 public void setAge(int age) {
				 this.age = age;
				 }
				}

JAVA MASTERY IN 4 DAYS: DETAILED TEACHING AGENDA

			Homework	Exercises on creating classes, objects, and methods

Day 3	Inheritance	Concept of Inheritance, Using the extends Keyword, Method Overriding	30 minutes	java
				public class Animal {
				 public void eat() {
				 System.out.println("This animal eats food");
				 }
				}
				public class Dog extends Animal {
				 public void bark() {
				 System.out.println("The dog barks");
				 }
				}
				Dog dog = new Dog();
				dog.eat();
				dog.bark();

	Polymorphism	Method Overloading (Compile-Time Polymorphism), Method Overriding (Runtime Polymorphism)	30 minutes	java
				public class MathOperations {
				 public int add(int a, int b) {
				 return a + b;
				 }
				 public double add(double a, double b) {
				 return a + b;
				 }
				}

	Abstract Classes and Interfaces	Abstract Classes and Methods, Implementing Interfaces	30 minutes	java
				public abstract class Animal {
				 public abstract void makeSound();
				}
				public class Dog extends Animal {
				 @Override
				 public void makeSound() {
				 System.out.println("Bark");
				 }
				}

	Exception Handling	Types of Exceptions, Try-Catch Blocks, Finally Block, Throw and Throws Keywords	30 minutes	java
				try {
				 int division = 10 / 0;
				} catch (ArithmeticException e) {
				 System.out.println("Division by zero is not allowed.");
				} finally {
				 System.out.println("This block is always executed.");
				}

			Homework	Create a simple project demonstrating inheritance and exception handling

Day 4	Collections Framework	Introduction to Collections, List, Set, and Map Interfaces, ArrayList and HashMap	30 minutes	java
				import java.util.ArrayList;
				public class Main {
				 public static void main(String[] args) {
				 ArrayList<String> list = new ArrayList<>();
				 list.add("Apple");
				 list.add("Banana");
				 list.add("Cherry");
				 for (String fruit : list) {
				 System.out.println(fruit);
				 }
				 }
				}

	File I/O Basics	Reading from and Writing to Files, Using FileReader and FileWriter	30 minutes	java
				import java.io.BufferedReader;
				import java.io.FileReader;
				import java.io.IOException;
				public class FileReadExample {
				 public static void main(String[] args) {
				 try (BufferedReader br = new BufferedReader(new FileReader("file.txt"))) {
				 String line;
				 while ((line = br.readLine()) != null) {
				 System.out.println(line);
				 }
				 } catch (IOException e) {
				 e.printStackTrace();
				 }
				 }
				}

	Final Project Discussion	Outline a Small Project Using Learned Concepts, Divide Project into Manageable Tasks	30 minutes	
	Project Work and Q&A	Start Working on the Project with Guidance, Address Any Questions or Issues	30 minutes	
			Homework	Complete the final project and prepare for a presentation or code review