JAVA MASTERY IN 4 DAYS: DETAILED TEACHING AGENDA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sDay** | **Topic** | **Sub-Topics** | **Time** | **Code Examples / Homework** |
| **Day 1** | **Introducti on to Java** | **Overview of Java, Features and Benefits, Setting up the**  **Java Development**  **Environment** | **20 minute**  **s** |  |
|  | **Basic Syntax** | **Writing a Simple Java**  **Program, Explanation of**  **Main Method, Compiling and**  **Running a Java Program** | **30 minute**  **s** | **java<br>public class HelloWorld {<br> public static void main(String[] args) {<br>**  **System.out.println("Hello, World!");<br> }<br>}<br>** |
|  | **Data**  **Types and**  **Variables** | **Primitive Data Types,**  **Declaring and Initializing**  **Variables, Type Casting** | **30 minute**  **s** | **java<br>int age = 25;<br>double salary =**  **50000.75;<br>char grade = 'A';<br>boolean isJavaFun = true;<br>** |
|  | **Control Statement**  **s** | **If-Else Statements, SwitchCase Statements, Loops (for, while, do-while), Break and**  **Continue Statements** | **40 minute**  **s** | **java<br>int number = 10;<br>if (number > 0) {<br> System.out.println("Positive number");<br>} else {<br>**  **System.out.println("Non-positive number");<br>}<br>** |
|  |  |  | **Home**  **work** | **Simple exercises to practice basic syntax and control statements** |
| **Day 2** | **Introducti on to**  **OOP** | **Concepts of Objects and Classes, Benefits of OOP** | **20 minute**  **s** |  |
|  | **Classes and Objects** | **Defining a Class, Creating Objects, Constructors** | **30 minute**  **s** | **java<br>public class Person {<br> String name;<br> int age;<br>}<br>Person person1 = new**  **Person();<br>person1.name = "John";<br>person1.age = 30;<br>** |
|  | **Methods** | **Defining and Calling**  **Methods, Method**  **Overloading** | **30 minut**  **es** | **java<br>public class Calculator {<br> public int add(int a, int b) {<br> return a + b;<br>**  **}<br>}<br>Calculator calc = new Calculator();<br>int sum = calc.add(5, 10);<br>System.out.println(sum);<br>** |
|  | **Encapsula**  **tion** | **Access Modifiers, Getters and Setters** | **40 minute**  **s** | **java<br>public class Person {<br> private String name;<br> private int age;<br> public String getName() {<br> return name;<br> }<br> public void setName(String name) {<br> this.name = name;<br> }<br> public int getAge() {<br> return age;<br> }<br> public void setAge(int age) {<br> this.age = age;<br> }<br>}<br>** |

1

JAVA MASTERY IN 4 DAYS: DETAILED TEACHING AGENDA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **Home**  **work** | **Exercises on creating classes, objects, and methods** |
| **Day 3** | **Inheritanc**  **e** | **Concept of Inheritance, Using the extends Keyword,**  **Method Overriding** | **30 minute**  **s** | **java<br>public class Animal {<br> public void eat() {<br> System.out.println("This animal eats food");<br> }<br>}<br>public class Dog extends Animal {<br> public void bark() {<br> System.out.println("The dog barks");<br> }<br>}<br>Dog dog = new Dog();<br>dog.eat();<br>dog.bark();<br>** |
|  | **Polymorp hism** | **Method Overloading**  **(Compile-Time**  **Polymorphism), Method**  **Overriding (Runtime**  **Polymorphism)** | **30 minute**  **s** | **java<br>public class MathOperations {<br> public int add(int a, int b) {<br> return a + b;<br> }<br> public double add(double a, double b) {<br> return a + b;<br> }<br>}<br>** |
|  | **Abstract Classes and**  **Interfaces** | **Abstract Classes and**  **Methods, Implementing**  **Interfaces** | **30 minute**  **s** | **java<br>public abstract class Animal {<br> public abstract void makeSound();<br>}<br>public class Dog extends Animal {<br> @Override<br> public void makeSound() {<br> System.out.println("Bark");<br> }<br>}<br>** |
|  | **Exception Handling** | **Types of Exceptions, TryCatch Blocks, Finally Block,**  **Throw and Throws Keywords** | **30 minute**  **s** | **java<br>try {<br> int division = 10 / 0;<br>} catch**  **(ArithmeticException e) {<br>**  **System.out.println("Division by zero is not**  **allowed.");<br>} finally {<br> System.out.println("This block is always executed.");<br>}<br>** |
|  |  |  | **Home**  **work** | **Create a simple project demonstrating inheritance and exception handling** |
| **Day 4** | **Collection**  **s**  **Framewor**  **k** | **Introduction to Collections,**  **List, Set, and Map Interfaces,**  **ArrayList and HashMap** | **30 minute**  **s** | **java<br>import java.util.ArrayList;<br>public class Main {<br> public static void main(String[] args) {<br> ArrayList<String> list = new ArrayList<>();<br> list.add("Apple");<br> list.add("Banana");<br>**  **list.add("Cherry");<br> for (String fruit : list) {<br>**  **System.out.println(fruit);<br> }<br> }<br>}<br>** |
|  | **File I/O Basics** | **Reading from and Writing to**  **Files, Using FileReader and**  **FileWriter** | **30 minute**  **s** | **java<br>import java.io.BufferedReader;<br>import java.io.FileReader;<br>import**  **java.io.IOException;<br>public class FileReadExample {<br> public static void main(String[] args) {<br> try**  **(BufferedReader br = new BufferedReader(new**  **FileReader("file.txt"))) {<br> String line;<br> while**  **((line = br.readLine()) != null) {<br>**  **System.out.println(line);<br> }<br> } catch**  **(IOException e) {<br> e.printStackTrace();<br> }<br> }<br>}<br>** |
|  | **Final**  **Project**  **Discussion** | **Outline a Small Project Using**  **Learned Concepts, Divide**  **Project into Manageable**  **Tasks** | **30 minute**  **s** |  |
|  | **Project**  **Work and**  **Q&A** | **Start Working on the Project with Guidance, Address Any Questions or Issues** | **30 minute**  **s** |  |
|  |  |  | **Home**  **work** | **Complete the final project and prepare for a presentation or code review** |

2