

Computer Programming in **JAVA**

Strengthened Senior High School Curriculum

JAKE RODRIGUEZ-POMPERADA
MAED-IT, MIT, PHD-TM

TABLE OF CONTENTS

PREFACE	III
ACKNOWLEDGEMENT	V
DEDICATION	VII
NOTE FROM THE AUTHOR	IX
CHAPTER I: INTRODUCTION TO COMPUTER PROGRAMMING IN JAVA	1 - 6
What is Computer Programming	
History and Evolution of Java	
Features and Advantages of Java	
Java in Schools, Business, and Industry	
Career Opportunities in Java	
Chapter I: Exercises: Quiz	7
Chapter I: Exercises: Essay	9
Chapter I: Exercises: Laboratory	11
CHAPTER II: SETTING UP THE JAVA DEVELOPMENT ENVIRONMENT	15 - 28
High School Java learners	
Installing Java on Windows	
Quick Summary (For SHS Students)	
Official Link to Download JDK	
Integrated Development Environments (IDE)	
Summary (For Students)	
Creating a Java Project	
Running a Java Program	
Chapter II: Exercises: Quiz	29
Chapter II: Exercises: Essay	31
Chapter II: Exercises: Laboratory	33
CHAPTER III: JAVA PROGRAM STRUCTURE AND SYNTAX	35 - 50
Basic Structure of a Java Program	

Class Declaration

Main Method

Complete Very Simple Java Program

The Main Method

Syntax of the Main Method

Explanation

Why the Main Method is Important

Data Types

Common Primitive Data Types and Examples

Variables and Constants

Declaring Variables in Java

Examples of Variables

What are Constants?

Declaring Constants in Java

Examples of Constants

Why Variables and Constants are Important

Comments and Coding Style

What are Comments?

Comments Help

Types of Comments in Java

Example Program with Comments

What is Coding Style?

Basic Coding Style Rules in Java

Good Coding Style Example

Why Comments and Coding Style are Important

Chapter III: Exercises: Quiz

51

Chapter III: Exercises: Essay

53

Chapter III: Exercises: Laboratory

55

CHAPTER IV: ALGORITHMS, FLOWCHARTS, AND PSEUDOCODE

59 - 83

Introduction to Algorithms

What is an Algorithm?

Simple idea

Brief History of Algorithms

Why Algorithms Are Important in Programming	
Relationship of Algorithms, Flowcharts, and Pseudocode	
Connection to Java Programming	
Types of Algorithms	
Relationship of Algorithms, Flowcharts, and Pseudocode	
Flowchart Symbols	
Flowchart Symbols and Their Role in Programming	
Relationship Summary	
Writing Pseudocode	
Relationship Between Algorithms, Flowcharts, and Pseudocode	
Why Pseudocode is Important for Java Programming	
Common Pseudocode Keywords	
Converting Logic to Java Code	
The Usual Programming Process	
How Flowcharts Help in Java Coding	
Chapter IV: Exercises: Quiz	85
Chapter IV: Exercises: Essay	87
Chapter IV: Exercises: Laboratory	89

CHAPTER V: JAVA OPERATORS AND INPUT/OUTPUT

93 - 116

Arithmetic Operators	
Common Arithmetic Operators in Java	
Sample Program 1: Addition of Two Numbers	
Sample Program Output	
Sample Program 2: Basic Arithmetic Operations	
Sample Program 3: Daily Allowance Calculation	
Relational and Logical Operators	
Logical Operators in Java	
Sample Program 1: Pass or Fail Checker	
Sample Program 2: Age Eligibility Checker	
Sample Program 3: Scholarship Qualification	
Sample Program 4: Attendance Check Using NOT Operator	
Assignment Operators	
Common Assignment Operators in Java	

Sample Program 1: Simple Assignment	
Sample Program 2: Using += Operator	
Sample Program 3: Daily Expense Calculator	
Sample Program 4: Multiplying and Dividing with Assignment	
User Input using Scanner	
The Scanner Class in Java	
Common Scanner methods	
Sample Program 1: Reading an Integer Input	
Sample Program 2: User Input with Arithmetic Operators	
Sample Program 3: Input with Relational and Logical Operators	
Sample Program 4: Reading Text Input	
Sample Program 5: Simple Input–Process–Output Program	
Output using Print Statements	
Print Statements in Java	
Sample Program 1: Using println()	
Sample Program 2: Using print()	
Sample Program 3: Output with Arithmetic Operators	
Sample Program 4: Output with User Input	
Sample Program 5: Displaying Pass or Fail Result	
Chapter V: Exercises: Quiz	117
Chapter V: Exercises: Essay	119
Chapter V: Exercises: Laboratory	121

CHAPTER VI: FLOW CONTROL STRUCTURES **125 - 138**

Conditional Statements	
Looping Statements	
Nested Loops	
Easy to Remember	
Multiplication Program Using Nested Loop in Java	
Break and Continue	
Common Programming Examples	
Chapter VI: Exercises: Quiz	139
Chapter VI: Exercises: Essay	141
Chapter VI: Exercises: Laboratory	143

CHAPTER VII: ARRAYS AND BASIC DATA STRUCTURES

147 - 167

Introduction to Arrays

Example 1: One-Dimensional Array

Example 2: Two-Dimensional Array

Importance of Arrays for Senior High School Students in Java Programming

One-Dimensional Arrays

Example 1: Declaring and Initializing a One-Dimensional Array

Example 2: Using a Loop to Display Array Elements

Example 3: One-Dimensional Array with User Input

Why One-Dimensional Arrays Are Important

Two-Dimensional Arrays

Example 1: Declaring and Initializing a Two-Dimensional Array

Example 2: Displaying All Elements Using Nested Loops

Example 3: Two-Dimensional Array with Student Grades

Importance of Two-Dimensional Arrays

Array Processing

Example 1: Displaying All Elements in an Array

Example 2: Finding the Sum and Average of Array Elements

Example 3: Finding the Largest Value in an Array

Example 4: Counting Even Numbers in an Array

Importance of Array Processing

Basic Sorting Concepts

Example 1: Sorting an Array Using Bubble Sort (Ascending Order)

Example 2: Sorting an Array Using Selection Sort

Example 3: Sorting Student Scores

Chapter VII: Exercises: Quiz

169

Chapter VII: Exercises: Essay

171

Chapter VII: Exercises: Laboratory

173

CHAPTER VIII: OBJECT-ORIENTED PROGRAMMING IN JAVA

177 - 204

Introduction to OOP

Classes and Objects

Example of a Class in Java

Example of Creating Objects in Java

Example Java Program that shows Classes and Objects	
Program Code Explanation	
Explanation of Main.java	
Why Do We Need Two Files?	
Encapsulation and Abstraction	
Encapsulation in Java	
Real-Life Example	
Java Example (Encapsulation)	
Why Encapsulation is Important	
Example Program that Demonstrates the Concept of Encapsulation	
Why Do We Separate the Code into Two Files?	
Encapsulation Works Better	
Easier to Maintain and Reuse	
Real-World Programming Style	
Java Rule for Main	
Abstraction in Java	
Real-Life Example	
Main.java — Line-by-Line Explanation	
Why Do We Separate the Code into Two Files?	
Reusability	
Professional Java Practice	
Java Rule for Running Programs	
Why Abstraction is Important	
Difference Between Encapsulation and Abstraction	
Inheritance	
Simple Idea of Inheritance	
Simple Example of Inheritance in Java	
Why Inheritance is Important	
Polymorphism	
Simple Idea of Polymorphism	
Why Polymorphism is Important	
Chapter VIII: Exercises: Quiz	205
Chapter VIII: Exercises: Essay	207
Chapter VIII: Exercises: Laboratory	209

Program Errors and Exceptions

What Are Program Errors?

What Is an Exception?

Exception Handling in Java (try-catch-finally)

Example (Handling division by zero)

File Processing Errors in Java

Example (Reading a file safely):

Why Exception Handling Is Important

Try-Catch-Finally

The try Block

The Catch Block

The Finally Block

Complete Example with File Processing

Why This Is Important

User-Defined Exceptions

Why Use User-Defined Exceptions?

How to Create a User-Defined Exception

Example 1: User-Defined Exception (Invalid Age)

User-Defined Exceptions in File Processing

Example 2: User-Defined Exception with File Processing

Key Points for Students

Reading Files

Reading Files in Java

Why Exception Handling is Important When Reading Files

Common Classes Used for Reading Files

Example 1: Reading a Text File Line by Line

Example 2: Reading a File with a User-Friendly Error Message

Example 3: Reading a File Safely Using finally

Key Points for Senior High School Students

Writing Files

Why Exception Handling is Important When Writing Files

Common Classes Used for Writing Files

Example 1: Writing Text to a File

Example 2: Appending Data to an Existing File

Example 3: Writing to a File Using PrintWriter

Example 4: Writing Files Safely Using finally

Key Points for Senior High School Students

Chapter IX: Exercises: Quiz 235

Chapter IX: Exercises: Essay 237

Chapter IX: Exercises: Laboratory 239

CHAPTER X: JAVA GUI, MULTITHREADING, AND PROJECT DEVELOPMENT 243 - 260

Introduction to Java GUI

Example: Simple Java GUI Window

Java GUI and Event Handling

Example: Button Click Event

Sample Program Output

Java GUI and Multithreading

Example: Simple Thread in a GUI

Java GUI and Project Development

Example Project Ideas for Students

Basic GUI Components

Common Basic GUI Components in Java

Basic GUI Components and Multithreading

Basic GUI Components in Project Development

Event Handling

Event Handling in Java GUI

Common GUI events

Event Handling and Multithreading

Why this is important

Event Handling in Project Development

Introduction to Multithreading

What is Multithreading?

Why Multithreading is Important in Java GUI?

Basic Multithreading Example in Java

Multithreading with Java GUI (Button Example)

What Happens Here?

Multithreading in Project Development	
Final Java Project	
Java GUI in the Final Project	
Simple GUI Example (Calculator Button)	
Multithreading in the Final Project	
Project Development Process	
Chapter X: Exercises: Quiz	261
Chapter X: Exercises: Essay	263
Chapter X: Exercises: Laboratory	265

REFERENCES

269