Exploring Scientific Computing with Java

A practical guide for logic and application building

Asst. Prof. Subhajit Adhikari

Prof. (Dr) Sunil Karforma

www.bpbonline.com

First Edition 2025

ISBN: 978-93-65897-593

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

Our beloved family:

Jayanta Kumar Adhikari *and* ***Rina Adhikari***

Nabanita Indra *and* ***Ishan Adhikari***

– Asst. Prof. Subhajit Adhikari

Late Radhapada Karforma *and* ***Late Smt Ranu Karforma***

– Prof. (Dr) Sunil Karforma

About the Authors

Asst. Prof. Subhajit Adhikari is presently working at the RCC Institute of Information Technology, Kolkata. He had worked in BSH Department in Institute of Engineering & Management, Salt Lake, Kolkata. He had also worked in the BCA department in DAITM Kolkata. He has more than 8 years of teaching experience. Additionally, he is a Registered PhD student of the Department of Computer Science at the University of Burdwan, West Bengal. Moreover, he did his MSC in Computer Science from St. Xavier's College, Kolkata and completed MTECH Software Engineering from MAKAUT(In-House), West Bengal. He was a gold medalist (First Class First) in MTECH Software Engineering in 2016. He qualified for UGC-NET and WBSET in 2018 and 2019 respectively and has published several research papers in the domain of network security and cryptography. He received the "Young Scientist Award" in the field of Computer Science in 2023 from Dr. Ram Avatar Shiksha Samiti (DRASS) registered under the Societies Act 21, 1860 (Reg. No. 265, 2007-08) since July 2007. He has been teaching "Java Programming" to undergraduate students for approximately 8 years. He also teaches "Artificial Intelligence" to the postgraduate students.

Prof. (Dr) Sunil Karforma has completed BE (Computer Science and Engineering) and ME (Computer Science and Engineering) from Jadavpur University. He has completed a Ph. D. in the field of Computer Science and he is presently holding the post of Dean (Science) between faculty and The University of Burdwan. Network security and e-commerce are his fields of research interest. He has published approximately 150 research papers in reputed National and International journals and proceedings. Also, he is the owner of 4 international patents and has successfully guided 12 PhD awardees.

About the Reviewer

Jason Alls is an accomplished author of three programming books. He has been programming computers since 1999, and has worked on many different systems for UK, Australasian, and American companies. He has also worked on call centre management softwares, debt management systems, reporting systems, advertising software, warehouse software, and educational assessment software.

Acknowledgement

I would like to express my deep gratitude to my family, whose unwavering support and encouragement have been a source of strength during the writing of this book. I especially extend my heartfelt thanks to my wife, Nabanita Indra, for her constant inspiration and understanding.

I also deeply appreciate the support of the Department of Computer Science, University of Burdwan as well as the Department of Computer Application in RCC Institute of Information Technology, Kolkata and CSE (AIML/CSBS) departments of IEM Salt Lake, Kolkata. Continuous encouragement has been helpful in bringing this work to life.

Special thanks are extended to BPB Publications for their guidance and expertise during the publishing process. The journey of revising this book has been a long and rewarding one, made possible by the invaluable participation and collaboration of reviewers, technical experts and editors.

I am grateful to my esteemed co-author and would like to acknowledge the important contribution of Prof. (Dr.) Sunil Karforma, whose guidance has been invaluable. My colleagues in the academic have also played a vital role, providing insights and feedback that have greatly enriched this work.

Finally, I would like to express my heartfelt thanks to all the readers who have shown interest in my book. Your support has been the driving force behind this effort, and I am truly grateful for your encouragement.

Preface

Building real-life applications is a complex task that requires a deep understanding of modern technologies and programming languages. The most powerful and widely used tools in application development are C++ and Java.

This book is designed as a comprehensive guide for anyone interested in mastering the art of argument construction using Java. It takes you on a journey from the fundamentals of Java programming to more advanced topics, including object-oriented programming and the practical application of Java technology to create robust, scalable, and efficient software solutions.

In this book, you will explore the key features of Java and learn how to leverage them to create applications that are not only efficient and dependable but also easy to maintain and extend. The book emphasizes best practices in logic generation and application development, offering a wealth of practical examples, diagrams, and illustrations to facilitate your understanding of complex concepts.

Whether you are a newbie eager to learn the basics of Java programming, or an experienced developer aiming to expand your expertise and improve your application development skills, there is something to offer in this book. It highlights design patterns and advanced logic-building strategies essential for developing real-life applications.

By the time you complete this book, you will have acquired the knowledge and skills necessary to become a proficient developer in Java application development. I hope you will find this book both informative and useful as you begin or continue your journey into the world of programming.

Chapter 1: Introduction to Java Programming - This chapter talks about Java's evolution and features and a comparative analysis of Java programming with C and C++. The details of a Java program syntax are outlined with the specifications required. This chapter also presents the characteristics essential for object-oriented programming with real-life examples. The relationship between Java Virtual Machine, Java Development Kit, and Java Runtime Environment is also discussed with diagrams with bytecode.

Chapter 2: Fundamental Building Blocks in Java - The concepts of data types of Javas, Java operators, Java expressions, conditional statements, and looping concepts are explained in this chapter. Let us first discuss the concept of a strongly typed language. Java is statically typed and a strongly typed language because, in Java, each type of data (such

as integer, character, hexadecimal, packed decimal, and so forth) is predefined as part of the programming language.

Chapter 3: Concept of Class, Objects, and Methods - This chapter combines examples from everyday life and illustrations to explain the concepts of classes, objects, and methods. The model, blueprint, or design is the class. In this chapter, the characteristics of classes and class types are shown. Introducing the idea of an object, which is a class instance. This chapter also illustrates the causes of object generation. To help with comprehension, the differences between classes and objects are listed. Next, the notion of methods is presented. In OOPs, they are referred to as the functions inside the class. In this case, the method call procedure is shown. The unique type of method with the same name as the class is called constructors. Examples of code are used to illustrate the constructor's idea. This chapter also covers this term and the crucial idea of method overloading.

Chapter 4: Concepts of Array, String and Vectors in Java - With sufficient knowledge of class, objects, methods, and constructors we are going to present some interesting and important topics, like array, string, vector, and wrapper class in this chapter. In different applications of mathematics, array is inseparable. String handling is also frequent in different real-life applications. String class methods like concat (), length() and substring() are very useful for manipulating different strings. In this chapter, we have presented a detailed explanation with coding examples of some of these methods. The StringBuffer class also explored with examples in this context. The class vector is also studied with diagrams for better understanding. The concept of auto-boxing and auto-unboxing with respect to wrapper class is included at the end of the chapter. After reading this chapter, students can learn from the basic to advanced level knowledge about the some of the most important programming constructs like array, String, vector, and wrapper class used for application development though java programming.

Chapter 5: Types of Inheritance in Java - In the previous chapter, the study of arrays, string and vectors has been discussed in detail. Now, we will focus on the concepts of inheritance and the need of inheritance along with the different types of inheritance. Then, we will cover the most important concept related to inheritance, its method overriding and how it is achieved with programs. Also, the usage of super keywords is added with different contexts. The functionality of the final variable, class and methods to restrict the subclass to inherit the properties and behaviors of the base class, will be studied in this chapter. The concept of interfaces will be presented to achieve multiple inheritance in java. Last, but not least, the different types of access modifiers are given with code and diagrams for better understanding of the learners.

Chapter 6: Concept of Thread in Java - In this chapter, the concept of multi-threading is placed in such a way that everyone can learn multi-threading to develop real life

applications. The thread life cycle is also presented in a diagram with a brief description. Different ways of creating thread in java are included with proper examples. The detailed discussion of thread class and runnable interface is one of the most important parts of this chapter along with thread priority. Important methods like join(), and isAlive() associated with thread programming have been elaborated in this chapter. Last but not least, the concepts of thread synchronization and inter thread communication have been added to this chapter.

Chapter 7: Exception Handling in Java - Every day, everyone faces some unexpected events, in programming, this is known as an exception. When an exception has occurred, the most important thing is to describe the necessary information about the exception. Next, we have to overcome the exceptional event, in programming, it is known as exception handling. In this chapter, if a program code generates one exception, the way to handle it is described in detail. In this regard, the term error is defined first and then how it can lead to generate exception is presented. The different types of errors are discussed. Then, exception is defined and how exception handling can be done in java is elaborated using programs.

Chapter 8: Package Creation in Java - Packages are an essential notion in Java. Java uses packages to eliminate name conflicts and manage access to classes, interfaces, and enumerations. The Java platform has an extensive class library (a collection of packages) that may be used in your own applications. This library is called the Application Programming Interface (API). Its packages cover the most typical tasks connected with programming for general use. A String object, for example, consists of state and behavior for character strings; a File object enables a programmer to easily create, delete, inspect, compare, or modify a file on the file system; a Socket object enables the creation and use of network sockets; and various GUI objects control buttons, check boxes, and other aspects of graphical user interfaces. There are literally hundreds of courses to pick from. This enables you, the programmer, to concentrate on the design of your specific application rather than the infrastructure needed to make it work.

Chapter 9: Stream and File Handling in Java - In Java, a stream is a series of data that is ordered. Java offers streams as a broad method of handling data I/O. To enable input and output operations in Java, the Java I/O system links a stream to a physical layer. In this chapter, we are going to present the detailed concept of stream and its types with pictorial representation. Then different classes and methods of different stream classes will be discussed. Also, the file handling in java is presented in this chapter. The relationship between streams and files are elaborated in this context.

Chapter 10: Applet Program in Java - Applet is a small program in java. It is not a standalone program. To implement the applet program, we need browser support. It has its own life

cycle. There are several methods present inside applet class to implement applet lifecycle. In this chapter, we are elaborating the building blocks of the applet programming using java. To give clear pictures of each concept, we have taken simple programs. The html programming is also taken into consideration to run applet programs. There is several programming examples are written in this chapter to demonstrate the utility and usability of applet programming building real life applications.

Chapter 11: Miscellaneous Programs Using Java- In this chapter, first we have provided the source code of two of the most important data structures Stack and Queue. Then, the Tic-tac-toe game is demonstrated using java program. Later, the two important symmetric key and asymmetric key algorithm AES and RSA have been implemented using java programming for encryption and decryption. At the end, the socket programming has been added in form of message passing between two programs, chat application and file sending between client and server.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/ug3je4m

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Exploring-Scientific-Computing-with-Java.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

Chapter 1
Introduction to Java Programming

Introduction

This chapter talks about Java's evolution and features and a comparative analysis of Java programming with C and C++. The details of a Java program syntax are outlined with the specifications required. This chapter also presents the characteristics essential for object-oriented programming with real-life examples. The relationship between **Java Virtual Machine** (**JVM**), **Java Development Kit** (**JDK**), and **Java Runtime Environment** (**JRE**) is also discussed with diagrams with byte code.

Structure

The chapter covers the following topics:

- Introducing Java
- History of Java
- Features of Java
- Relationship between JVM, JRE, and JDK
- Bytecode

Objectives

This chapter aims to cover the history of Java programming, tracing its origins through a programming language family tree. It also explores the shift from procedural to object-oriented programming, highlighting reasons for the transition. The chapter explains the differences between C, C++, and Java programming constructs, emphasizing Java's features in an object-oriented context with examples. Additionally, it explores the architecture of the Java Virtual Machine, focusing on byte code operations. The goal is to provide readers with a comprehensive understanding of Java's evolution, principles, and runtime environment.

Introducing Java

Everyone uses language as a mode of communication to spread ideas and arguments to others regularly. A programming language created artificially directs how a machine, especially a computer, behaves. Syntactic (the syntax of a program code) and semantic (meaning of a program code) principles are used to design programming languages, just as human languages are used to specify their structure and meaning, respectively. Programming languages are used to define algorithms accurately and ease communication about organizing and manipulating information. The goal of a programming language is to make it possible for people to provide instructions to computers, allowing the machines to behave in a predetermined and desired way. The family tree of programming languages is shown in *Figure 1.1:*

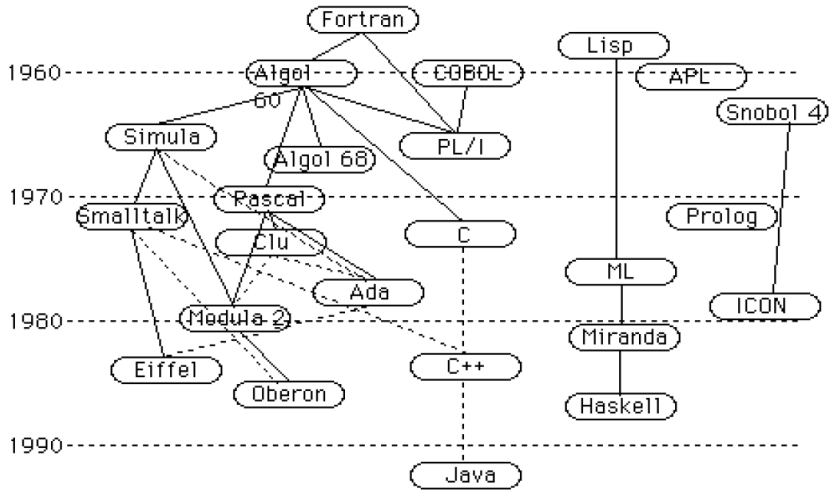

Figure 1.1: *Family tree of important programming languages*

Now consider the most trending languages in 2022. This is shown as per the job requirement in *Figure 1.2*. Java is the second most used programming language according to web application development.

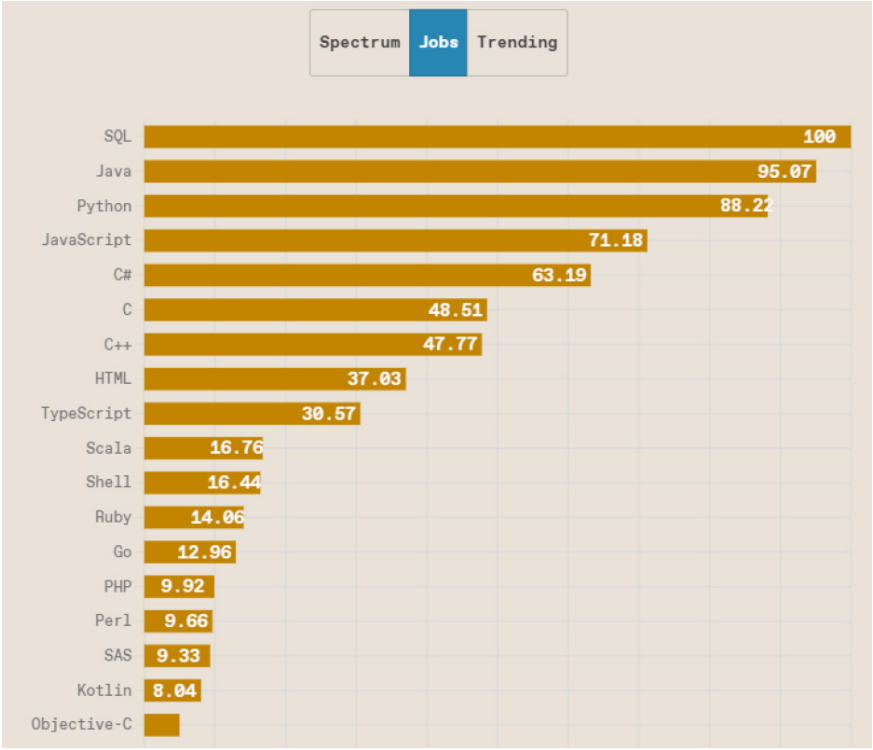

Figure 1.2: *Top programming languages 2022*

Java is a general-purpose language, often referred to as a strongly typed language. It is commonly used for object-oriented programming. Java program codes are typically compiled to the bytecode instruction set and binary representation defined in the JVM specification. The program codes written in Java can be categorized as *Applications*, *Apps*, *Applets*, or *Servlets*.

Java programs, which run locally and independently on a computer, are called **applications**. Applications and apps are similar, although the term *app* is usually used to describe software that runs on mobile devices, such as mobile phones. Applets are Java programs downloaded over the internet and run by a JVM, which is normally included with the browser.

The Servlet programs are Java programs created for a web server to run and give functionality to websites. Even in modern web development, this type of Java program is frequently used. In the next section, we will briefly discuss the history of Java.

History of Java

Sun Microsystems created Java in the early-to-mid 1990s, and the development continues to this day. A team headed by *James Gosling* set up to work on a programming language for consumer electronic devices, and the language came to be known as **Oak,** which was then renamed to Java. Here are some of the more significant years in the Java language's early history in *Figure 1.3*:

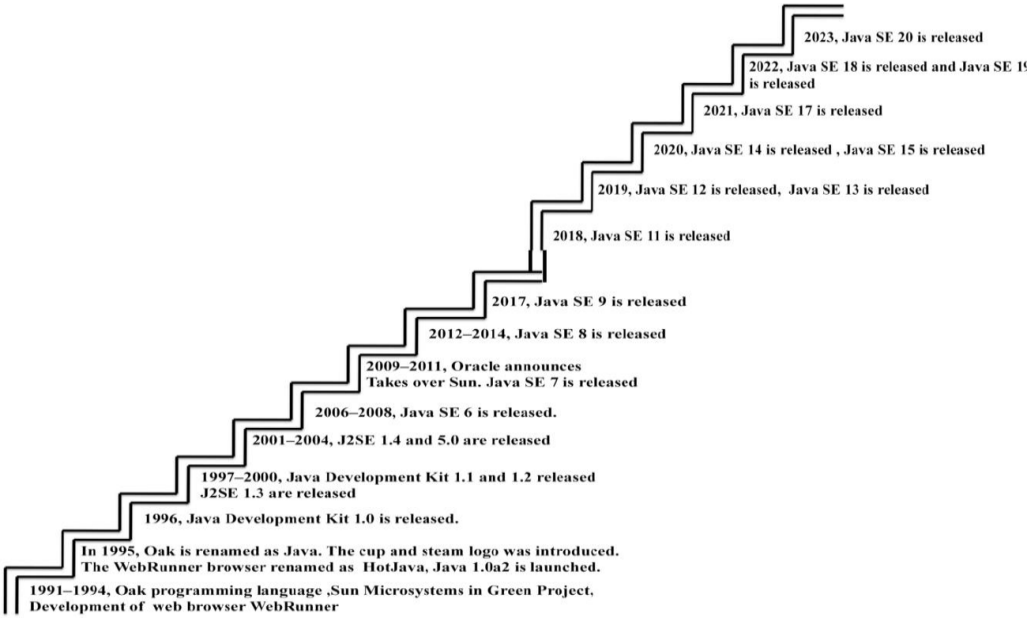

Figure 1.3: *History of Java*

Transition from procedural programming to OOP

There are several reasons for adopting **Object-Oriented Programming** (**OOP**) rather than procedural programming. Some of them are as follows:

- The data of a system and the operations that alter the data are often kept separate in procedural programming. The main benefit of object-oriented programming is that the data and the operations that alter the data or the code are both contained within the object. As opposed to procedural or structured design, where attributes and behaviors are often separated, object-oriented design integrates characteristics and behaviors into a single object.
- Comparing an object-oriented model to procedural programming, we can say that in terms of proper design, there is no such thing as global data. In object-oriented systems, this characteristic guarantees a high level of data integrity. The terms *attributes* and *methods* are used in object-oriented programming to describe data and behaviors. Data hiding refers to the practice of restricting access to specific properties and/or methods.

Let us examine how procedural and object-oriented programming languages vary from one another, as shown in *Table 1.1*:

Procedural	OOP
Programming is done based on task or procedure	Object is used rather than tasks
A task is known as a small part of a problem.	Object represents a person, a table or a clock. It has functions and characteristics
It follows Top-down approach.	It follows Bottom-Up approach.
It does not support access modifiers.	It supports access modifiers
It is not easy to add new data and functions.	It is simple to add new data and functions.
It is less secure since it lacks a suitable method of hiding data.	It has the concept of data hiding so it is more secure.
The importance of function is more than data.	The importance of data is more important than function.
The concept of inheritance is not supported.	The concept of inheritance is supported in three modes: public, private & protected.
Example: C, , Fortran, VB, and Pascal.	Example- Simula, Smalltalk, C++ and Java

Table 1.1: *Comparison between procedural and OOP approach*

Comparison between C, C++ and Java programming

There are several differences between C, C++, and Java language, some of which are given in *Table 1.2*:

C	C++	Java
C can only be compiled, not interpreted	C ++can only be compiled, not interpreted	Java language can be compiled and interpreted.
C supports both Pass by value and Pass by reference.	C++ supports both Pass by value and Pass by reference	Java supports only Pass by value technique
C does not support both operator and method overloading	C++ supports both operator and method overloading	Java supports only method overloading
C supports **`goto`** statement	C++ supports **`goto`** statement	Java does not support the **`goto`** statement
There is no concept of Inheritance in C.	In C++, both single inheritance and multiple inheritances are supported	Java supports the single inheritance, multilevel inheritance and hierarchical heritance using the extend keyword.