# A Quant’s Learning Path in Computer Science

Version 1.0, 2025-01-14

# Four Building Blocks

- A core programming language: C/C++/Python/Java; C/C++ preferred
- Data Structure and Algorithms
- Computer Architecture / Hardware
- Operating System
- “Glues”

# Logic behind the building blocks

- You describe the solution to a problem in algorithm
- Each algorithm works with certain data structures. E.g. a sorting algorithm with an array of integers: tradeoff between memory and time
- You need a programming language to implement your algorithm
- For efficient implementation of algorithms, you need knowledge of hardware to understand how code is executed at machine level. E.g. memory allocation
- When many task are working simultaneously, you need a good scheduler – operating system

# But you also need “glues” to smooth your work

- Source code editing, compilation, and linking: IDE (integrated development environment)
  - Visual Studio for C/C++
  - PyCharm for Python
  - Eclipse for Java
- Version control system to track your code change: Git
- File management: Linux, bash shell, awk, sed, etc.

# Programming Languages: Overview

- Recommend the C/C++ combo for hard core programming
  - C works directly with operating system, giving you intuition of code execution at machine level
  - C++ is the mother of all object oriented programming languages: once you master C++, Java/Python etc. are easy
- Python is ideal for application due to its rich ecosystem: scientific computation, machine learning, etc.
- Java also has rich ecosystem, widely used in system building

# Programming Languages: C/C++

- Brian W. Kernighan & Dennis M. Ritchie. *The C Programming Language*, 2nd edition. Prentice Hall, 1988.
- C++
  - Fast tutorial: https://cplusplus.com/doc/tutorial/
  - Stephen Prata. *C++ Primer Plus*, 5th edition. SAMS, 2004.

**C/C++**

- Yale N. Patt & Sanjay J. Patel. *Introduction to Computing Systems: From Bits and Gates to C and Beyond*, 2nd edition. McGraw-Hill Education, 2003.
- Brian W. Kernighan & Dennis M. Ritchie. *The C Programming Language*, 2nd edition. Prentice Hall, 1988.
- Cplusplus.com. C++ Tutorials.
- Paul Deitel. *C++ Fundamentals I and II LiveLessons*. Prentice Hall, 2010. Safari Books Online.
- Stephen Prata. *C++ Primer Plus*, 5th edition. SAMS, 2004.
- Stanley B. Lippman, Josee Lajoie, & Barbara E. Moo. *C++ Primer*, 5th edition. Addison-Wesley Professional, 2012.
- Stephen C. Dewhurst. *C++ Common Knowledge*. Addison-Wesley Professional, 2005.
- Stanley B. Lippman. *Essential C++*. Addison-Wesley Professional, 1999.
- Scott Meyers. *Effective C++: 55 Specific Ways to Improve Your Programs and Designs*, 3rd edition. Addison-Wesley Professional, 2005.
- Isocpp.org. C++ FAQ.
- Heard at JPMC (2015): C++ Coding Guidelines, C++ Coding Standards.

# Programming Languages: Java

- Cay S. Horstmann. *Core Java for the Impatient (Covers Java SE 8)*. Addison-Wesley Professional, 2015.
- Joshua Bloch. *Effective Java, 3rd edition*. Addison-Wesley Professional, 2018.
- Dheeru Mundluru. *Java In-Depth: Becoming a Complete Java Engineer*. Last updated 10/2018, Udemy. [very deep dive!!!]
- Maven as a project management tool:
  - Sonatype. 1) *Maven by Example*. 2) *Maven: The Complete Reference*.
  - Balaji Varanasi & Sudha Belida. *Introducing Maven*. Apress, 2014.
  - Raghuram Bharathan. *Apache Maven Cookbook*. Packt Publishing, 2015.
  - Jason Taylor. *Maven Crash Course: Step-by-Step Introduction for Beginners*. Udemy.
  - Maven Central Repository. 1) Repo. 2) Search Engine. 3) Nexus Repository Manager.
  - Apache Maven Project. 1) Maven Documentation. 2) Frequently Asked Technical Questions. 3) Apache Maven Javadoc Plugin. 4) Books and Resources.
  - The Maven 2 POM demystified.
  - Maven Quick Reference Card.
  - Sample pom.xml.

# Programming Language: Python

- Jake VanderPlas. *A Whirlwind Tour of Python*. O'Reilly Media, 2016.
- Jake VanderPlas. *Python Data Science Handbook*. O'Reilly Media, 2016.
- IDE: PyDev, Spyder, PyCharm, Visual Studio.
- Luciano Ramalho. *Fluent Python*. O'Reilly Media, 2015. [very deep dive!!!]
- Adnan Aziz, Tsung-Hsien Lee, and Amit Prakash. *Elements of Programming Interviews in Python: The Insider's Guide*. 2016 [the ultimate interview preparation!!!]

# Hardware/Computer Architecture

- Yale N. Patt & Sanjay J. Patel. *Introduction to Computing Systems: From Bits and Gates to C and Beyond*, 2nd edition. McGraw-Hill Education, 2003.
  - First half is a mini course in electrical engineering: logical gates, circuits, machine code, assembly language, memory stack
  - Second half is a course on C programming language
- Jon Stokes. *Inside the Machine: An Illustrated Introduction to Microprocessors and Computer Architecture*. 2007

# Operating System

- Have forgotten what I learned; understood the vocabulary for communication
- (The dragon book) Silberschatz, Galvin and Gagne. *Operating System Concepts Essentials*. 2011.

# Data Structure and Algorithm

- 张乃孝：《算法与数据结构：C语言描述》
- Adnan Aziz, Tsung-Hsien Lee, and Amit Prakash. *Elements of Programming Interviews in Python: The Insider's Guide*. 2016
- HackerRank: https://www.hackerrank.com/
- Best way to learn data structure and algorithm is implement all the classical algorithms, e.g. as explained in 张乃孝
- Then upload your code to GitHub as part of your job application portfolio

# “Glues”

- Don’t learn from big books; learn from short, quick tutorials
- Udemy provides a wide variety of useful short course
- Unix Programming Tools: http://cslibrary.stanford.edu/107/
- Chris Johnson. *Pro Bash Programming: Scripting the Linux Shell*. Apress, 2009.
- Daniel J. Barrett. *Linux Pocket Guide*, 2nd Edition. O'Reilly, 2012.
- Udemy Free course: Command Line Essentials Git Bash for Windows
- DevOps: https://www.geeksforgeeks.org/devops-tutorial/

# More: design patterns, software architecture, and legacy code

- Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994. [*Gang of Four*]
- TutorialsPoint.com. Design Patterns in Java Tutorial.
- Oodesign.com: Design Patterns.
- HowTODoInJava: GoF Design Patterns.
- Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 2017.
- Stanford University. Programming Paradigms (by Jerry Cain, 2008). YouTube.
- The key points of The Legacy Code Programmer's Toolbox
- The key points of Working Effectively with Legacy Code