

Table of Contents

Computer Programming

Disciplines & Methodologies

Tools & Technologies

Links

3

5

6

8

Computer Programming

Computer programming is the process of writing and testing code that enables applications and software programs to operate successfully. Computer programmers use specialized languages to communicate with computers, applications and other systems to perform a specific task or solve a problem. Computer programming can be simple or complex depending on the requirements and goals of the program.

Some examples of computer programming are creating websites, games, mobile apps, operating systems and artificial intelligence.

Programming Languages Full Stacks (Software Bundles)

- [Software design pattern](#)
- [Data Structures](#)
- [Web Technologies](#)
- [Computer Networking](#)
- [Cyber Security](#)
- [Development](#)

Computer programming involves writing code in programming languages such as [Java](#), [Python](#), or [C++](#), to create software applications. Computer programming requires a deep understanding of programming languages, algorithms, data structures, and software development principles.

[Citizen Development](#), on the other hand, involves using low-code or no-code platforms to create software applications. Citizen developers are typically business users who have limited technical knowledge but can use low-code platforms to develop software applications using drag-and-drop interfaces and pre-built templates. Citizen development enables non-technical users to develop software applications quickly and efficiently, without requiring extensive technical expertise.

[Low-code development](#) is similar to citizen development but typically involves more technical knowledge and expertise. Low-code platforms provide a visual development environment where developers can build software applications using pre-built components and drag-and-drop interfaces. Low-code development is faster and more efficient than traditional computer programming, but it still requires some technical knowledge and expertise to use effectively.

Computer programming is a traditional approach to software development that requires deep technical knowledge and expertise, while citizen development and low-code development are newer approaches that allow non-technical users to develop software applications quickly and efficiently using low-code or no-code platforms. Each approach has its advantages and disadvantages, and the best approach will depend on the specific needs and requirements of the software development project.

What is computer programming?

Computer programming is the process of designing, writing, testing, and maintaining software programs using [Programming Languages](#).

Why is computer programming important?

Computer programming is important because it enables the creation of software applications and systems that are used in various industries and sectors, such as finance, healthcare, education, and entertainment.

What are some common programming languages?

Some common programming languages include Java, Python, C++, JavaScript, Ruby, and Swift.

What are some common programming paradigms?

Some common programming paradigms include imperative, functional, object-oriented, and procedural programming.

What are some common tools and technologies used in computer programming?

Common tools and technologies used in computer programming include integrated development environments (IDEs), version control systems (such as Git), application programming interfaces (APIs), and software development kits (SDKs).

What is the difference between a compiled and interpreted programming language?

A compiled programming language is one in which the source code is translated into machine code before the program is run. An interpreted programming language is one in which the source code is translated into machine code at runtime.

What is debugging in computer programming?

Debugging is the process of identifying and fixing errors, or bugs, in a program. This is typically done using debugging tools and techniques, such as stepping through the code, setting breakpoints, and inspecting variables.

How can computer programming be used in data analysis?

Computer programming can be used in data analysis by using programming languages and libraries to manipulate, visualize, and analyze data. Examples of programming languages used in data analysis include Python, R, and SQL.

How can computer programming be used in web development?

Computer programming can be used in web development by using programming languages, such as HTML, CSS, and JavaScript, to create and design web pages and web applications.

How can computer programming be used in artificial intelligence and machine learning?

Computer programming can be used in artificial intelligence and machine learning by using programming languages and libraries, such as Python and TensorFlow, to build and train machine learning models, and to develop AI algorithms and applications.

Snippet from [Wikipedia](#): **Computer programming**

Computer programming or **coding** is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.

Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. While these are sometimes considered programming, often the term *software development* is used for this larger overall process – with the terms *programming*, *implementation*, and *coding* reserved for the writing and editing of code per se. Sometimes software development is known as *software engineering*, especially when it employs formal methods or follows an engineering design process.

[Creative Commons Attribution-Share Alike 4.0](#)

Disciplines & Methodologies

- [Agile Unified Process](#)
- [Build automation](#)
- [Container Management](#)
- [Cross-platform](#)
- [Cyber Security](#)
- [DevOps toolchain](#)
- [Disciplined agile delivery](#)
- [Dynamic systems development method](#)
- [Extreme programming \(XP\)](#)
- [Feature toggle](#)

- [GitOps](#)
- [Information security](#)
- [Installation](#)
- [Integrated Development Environment \(IDE\)](#)
- [Machine learning \(ML\)](#)
- [Object-oriented programming](#)
- [Pair Programming](#)
- [Profiling](#)
- [Programming language](#)
- [Repository Management](#)
- [Robotic Process Automation \(RPA\)](#)
- [SDLC](#)
- [Software build](#)
- [Software deployment](#)
- [Software development process](#)
- [Software documentation](#)
- [Software Engineering Management](#)
- [Software prototyping](#)
- [Static program analysis](#)
- [System integration](#)
- [Unit testing](#)

Tools & Technologies

- [Angular](#)
- [Apache ANT](#)
- [Apache Buildr](#)
- [Apache Kafka](#)
- [Apache Subversion](#)
- [Atom](#)
- [AWS Cloud9](#)
- [Azure DevOps](#)
- [BitBucket](#)
- [Broccoli](#)
- [BuildMaster](#)
- [ChatGPT](#)
- [CMake](#)
- [CodeBeautify](#)
- [CodeLobster](#)
- [Codenvy](#)
- [Docker](#)
- [Eclipse](#)
- [Elasticsearch](#)
- [Electron](#)
- [Gerrit Code Review](#)
- [GitHub](#)
- [GitHub Copilot](#)
- [GitHub Learning Lab](#)
- [GitLab](#)
- [Gradle](#)

- Grunt
- Gulp
- HP ALM
- IntelliJ IDEA
- LaunchDarkly
- Linx
- Logscape
- Lucene
- LuntBuild
- MariaDB
- Meister
- Microsoft Graph
- Microsoft Graph Toolkit
- Microsoft Visual Studio
- MSSQL
- MySQL
- NAnt
- neo4j
- NetBeans
- Node.js
- Notepad++
- Octopus Deploy
- Oracle DB
- Postman
- PowerShell
- PyTorch
- QuickBuild
- Rake
- Rational solution for CLM
- Rational Team Concert
- sbt
- SCons
- Selenium
- SonarQube
- SQLite
- Stackify
- StarTeam
- SVG.js
- Team Foundation Server
- TeamForge
- Test automation
- UrbanCode Build
- Visual Build
- Visual Studio
- VSALM
- Visual Studio Code
- Visual Studio Team Services
- Vue.js
- Webpack
- WebView2
- Xamarin

- [Yeoman](#)

Links

- [AlgoExpert](#)
- [Bento](#)
- [BitDegree](#)
- [Code Avengers](#)
- [Code Conquest](#)
- [Code.org](#)
- [Code4Startup](#)
- [codeasy.net](#)
- [Codebyte](#)
- [Codecademy](#)
- [CodeGym](#)
- [CodeHS](#)
- [Codementor](#)
- [Codewars](#)
- [Coding Dojo](#)
- [CodingBat](#)
- [Codingem.com](#)
- [Comparison of open-source configuration management software](#)
- [DEV Community](#)
- [DevDocs](#)
- [Devhints](#)
- [Devicon](#)
- [Edabit](#)
- [freeCodeCamp](#)
- [GeeksforGeeks](#)
- [Google Developers Learn](#)
- [Guru99](#)
- [HackerNoon](#)
- [HackerRank](#)
- [hackr.io](#)
- [Khan Academy](#)
- [Learn to Code \(GitHub Collection\)](#)
- [Learn X in Y minutes](#)
- [List of ECMAScript engines](#)
- [List of Freely Available Programming Books](#)
- [MDN Web Docs](#)
- [MIT OpenCourseWare](#)
- [One Month](#)
- [Programiz](#)
- [Scratch](#)
- [SitePoint](#)
- [Sololearn](#)
- [Stack Overflow](#)
- [Studytonight](#)
- [The Odin Project](#)
- [Thinkful](#)

- [Tutorials Point](#)
- [Unzip.dev](#)
- [Upskill](#)
- [w3Schools](#)
- [Ajax](#)
- [Algorithm](#)
- [Change control](#)
- [Chaos model](#)
- [Class](#)
- [Compiler](#)
- [CUDA](#)
- [Debugger](#)
- [Document Object Model \(DOM\)](#)
- [Don't repeat yourself \(DRY\)](#)
- [Enterprise service bus \(ESB\)](#)
- [Feature Flags](#)
- [Git](#)
- [Graphical user interface](#)
- [GRASP](#)
- [Inner Source](#)
- [JSON](#)
- [KISS](#)
- [Library](#)
- [Microservices](#)
- [MSBuild](#)
- [.NET](#)
- [.NET MAUI](#)
- [NLP](#)
- [OData](#)
- [Open Source](#)
- [Package manager](#)
- [Progressive web application \(PWA\)](#)
- [Refactoring](#)
- [Representational State Transfer \(REST\)](#)
- [REST](#)
- [Scalable Vector Graphic \(SVG\)](#)
- [SDK](#)
- [Service-provider network](#)
- [Software design pattern](#)
- [Software framework](#)
- [Software versioning](#)
- [SOLID](#)
- [Source-to-source compiler](#)
- [SWEBOK](#)
- [TFVC](#)
- [Version control](#)
- [WAN](#)
- [XAML](#)
- [XML](#)
- [You aren't gonna need it](#)

development, skill

ToDo



-
- Object-Oriented Programming (OOP)
- Functional Programming (FP)
- Procedural Programming
- Algorithm Design and Analysis
- Data Structures and Algorithms
- Software Development Life Cycle (SDLC)
- Agile Development and Scrum Methodology
- DevOps and Continuous Integration/Continuous Delivery (CI/CD)
- Version Control and Source Code Management (SCM)
- Test-Driven Development (TDD) and Unit Testing
- Debugging and Troubleshooting
- Application Programming Interfaces (APIs)
- Web Development and Front-End Technologies (HTML, CSS, JavaScript)
- Back-End Technologies (Node.js, Java, Python)
- Database Management and SQL
- Mobile App Development (iOS, Android)
- Game Development
- Artificial Intelligence (AI) and Machine Learning (ML)
- Computer Graphics and Visualization

From:

<https://www.almbok.com/> - **ALMBoK.com**

Permanent link:

<https://www.almbok.com/programming/programming>

Last update: **2023/05/01 17:53**

