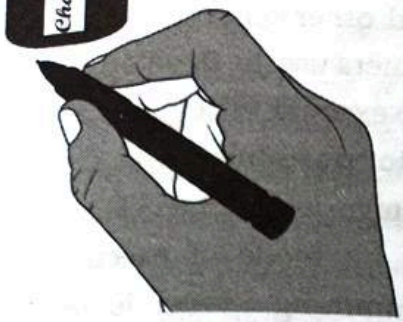
Chapter 8



CODING

CHAPTER OUTLINE

After studying this chapter, the reader will be able to understand the

- Programming Languages
- Program Development Tools

PROGRAMMING LANGUAGES

A programming language is a set of commands, instructions, and other syntax use to create a computer program. It is a computer language that the programmers use to develop software programs, scripts, or other sets of instructions for computers to execute. They are a formal language that specifies a set of instructions for computer to do something. Generally programming languages can be classified by various different categories such as classification by programming model, classification by typing, classification by mode of execution and classification by modularity. However, classification by programming model is the most popular one. Programming languages are typically classified based on the paradigm that they support. The following are the main programming models or classes now in general use and they are used to express a computation nowadays:

- Imperative programming languages
- Functional programming languages
- Object-oriented programming languages
- Logical programming languages

Imperative Programming Languages

In this class of programming languages, programs are usually decomposed into computation steps (such as instructions, statements or commands), reflecting the step-wise execution of programs in traditional hardware. Procedures or routines, which are also called sub-programs, are used to modularise the program. Imperative programs provide the accurate descriptions of "How to solve a given problem". **FORTRAN, ALGOL 60, C** and **PASCAL** are examples of imperative programming languages

Functional Programming Languages

Programs in this class are functions that can be composed to create new functions similar to building functions in the mathematical theory of functions. Functional languages are also called declarative languages because this class of language is more focus on what should be computed in the program rather than focusing on how the program should be computed. Declarative languages highlight the use of expressions, which are evaluated by simplification. Examples of functional languages are ***Lisp***(List Processor) , ***CLOS*** (Common Lisp Object System), **Miranda**, and **Goferas** subset of **Haskell** language, Erlang language **SML** (Standard ML language), *Scheme* was popular because of its clean language design and **Caml**

Object-Oriented Programming Languages

Object-oriented programs are a set of objects that can be accessed via the defined methods or operations on them which are organised in a hierarchy way. An obvious example of object-oriented language is **C++,Java ,PHP,C#.** Object-oriented design focuses on the combination of fields/data and methods/operations, which are called entities. Some researches show that object-orientation is a general approach for programming rather than a specific type that is easily classifiable. Thus, sometimes object-orientation can be considered as a feature of imperative languages. However, it can also be found in functional languages and combined with logic languages as

Logical Programming Languages

This class of languages defines a problem that required to be solved rather than describing an algorithmic implementation. The main focus is on the specification of the problem that needs to be solved instead of declaring the way in which it is solved. Logical languages are also declarative. The most popular logical programming language is Prolog, which was designed in 1970 by Comerauer, Kowalski and Roussel. Nowadays, logic programming is combined with constraint-solving and called modern logic programming languages

PROGRAM DEVELOPMENT TOOLS

These are a set of tools to help software developer in doing their tasks to create software product. Some of the development tools used these days are described with their features below:

1. **Linx**

   Linx is a low code IDE and server. IT pros use Linx to quickly create custom automated business processes, integrate applications, expose web services and to efficiently handle high workloads.

   - Easy-to-use, drag-and-drop interface
   - Over 100 pre-built functions and services for rapid development
   - One-click deployment to any local or remote Linx Server directly from the IDE
   - Input and outputs include nearly any SQL & NoSQL databases, numerous file formats (text and binary) or REST and SOAP Web services
   - Live debugging with step through logic
   - Automate backend processes via timer, directory events or message queue or expose web services, and call APIs via HTTP requests

2. **Buddy**

Buddy is a smart CI/CD tool for web developers designed to lower the entry threshold to DevOps. It uses delivery pipelines to build, test and deploy software. The pipelines are created with over 100 ready-to-use actions that can be arranged in any way – just like you build a house of bricks.

- 15-minute configuration in clear & telling UI/UX
- Lightning-fast deployments based on changesets
- Builds are run in isolated containers with cached dependencies
- Supports all popular languages, frameworks & task managers
- Dedicated roster of Docker/Kubernetes actions
- Integrates with AWS, Google, DigitalOcean, Azure, Shopify, WordPress & more
- Supports parallelism & YAML configuration

3. **HeadSpin**

HeadSpin is the world's first Connected Intelligence Platform that provides Web, Mobile, IoT, and 5G solutions to unify testing, monitoring, and analytics across applications, devices, and networks. HeadSpin empowers development, QA, operations, and product teams to optimize connected experiences and ensure digital business success.

Features

- Access to 300+ devices over 30+ countries on shared device cloud
- Remote Debugging
- Real SIM-enabled Android and iOS devices
- 500+ parallel tests
- Regression Testing
- Automation
- Localization Testing
- Locally debugging and code profiling

IDE (Integrated Development Environment)

4. **NetBeans**

NetBeans is a popular, Free, open-source IDE. It allows developing desktop, mobile and web applications.

Features:

- Support for fast & smart code editing
- Easy & Efficient Project Management process
- Rapid User Interface Development
- Helps to write bug-free code
- NetBeans IDE offers superior support for C/C++ and PHP developers
- It can be installed on any OS which supports Java, from Windows to Linux to Mac OS X systems

5. **Cloud9 IDE**

Cloud9 IDE is an online integrated software development environment. It supports many programming languages like C, C++, PHP, Ruby, Perl, Python, JavaScript and Node.js.

Features:

- Allows to clone entire development environment
- Built-In Terminal for command-line wizard
- Code Completion suggestions helps software developers to code faster and avoid typos
- The Debugger helps developers to set breakpoints, and inspect variables of any JS/Node.js app
- Simply drag any file or Terminal to create multiple split views
- Developers can select an extensive set of default Runners to execute app, such as Ruby, Python, PHP/Apache

6. **Zend Studio**

Zend Studio allows software developers to code faster, debug more easily. It is next-generation PHP IDE designed to create apps for boosting developers' productivity. It scales according to the DPI settings of the underlying operating system.

Features:

- Code faster with up to performance improvements in indexing, validation, searching for PHP code
- Offering debugging capabilities with Xdebug, Zend Debugger, and Z-Ray integration
- Extensive plugin provided by the large Eclipse eco-system
- It supports development tools including Docker and Git Flow
- Deploy PHP applications on any server for Amazon AWS and Microsoft Azure

7. **Atom**

Atom is a solid all-around text-editor. It is fully free and open source. It can be customized to do anything but without a need of modifying the config file.

Features:

- Atom works across many popular operating systems like OS X, Windows, or Linux
- It helps developers to write code faster with a smart, flexible auto complete
- Easily browse and open whole project or multiple projects in one window
- It is possible to split Atom interface into multiple panes to compare and edit code across files
- Find, preview, and replace text type in a file or across the entire project

8. **Spiralogics Application Architecture**

Spiralogics Application Architecture (SAA) is a cloud-based software development tool. It allows users to build and customize their applications online and deploy them. It also allows users to choose from a set of prebuilt applications or customize them it from scratch.

Features:

- Create customized pages
- Built-in HTML editor
- Interactive Dashboard builder
- Preview Changes before publishing the application
- Predefined processes like Save, Delete, Accept, Reject, and Email
- Allows customization of look and feel of page components
- Create custom processes not already defined

9. **CodeLobster**

Codelobster streamlines and simplifies PHP software development process. It supports CMS like Wordpress, Drupal, Joomla, and Magento.

Features:

- PHP, HTML, JavaScript, CSS code highlighting
- Autocomplete of tags, attributes for a current tag, closing tags.

- Inspector makes it simple to find HTML elements and their styles buried deep in the page
- Autocomplete of style property names and values
- It allows autocompleting of keywords, DOM elements, and their properties
- It offers PHP Advanced autocomplete

10. **CodeCharge Studio**

CodeCharge Studio offers the fastest way to build applications. This tool helps to develop data-driven Web sites or enterprise Internet and Intranet systems.

Features:

- Avoid costly errors and misspellings by generating consistent, well-structured code
- Eliminate time-consuming programming tasks and build scalable, robust Web Applications
- Helps to convert any database into a web application in very less amount of time
- Analyze and modify generated code to learn web technologies and take on programming projects in any environment

Frameworks

11. **Bootstrap**

Bootstrap is a responsive framework for developing with HTML, CSS, and JS. It has many in-builds components, which you can easily drag and drop to assemble responsive web pages.

Feature:

- Bootstrap enables utilization of ready- made blocks of code
- It ensures consistency irrespective of who's working on the project
- It offers extensive list of components
- Base Styling for most HTML Elements
- Bootstrap can be customized according to the specific need of the project

12. **Expression Studio**

Expression Studio is a set of a family of tools for professional designer's developers. It is a robust professional design tool which gives creative freedom to developers.

Features:

- It revolutionizes the speed of prototyping
- It allows creation of effective UI with sample data
- Fast, flexible, seamless workflow helps to excel the entire development process
- These tools save time of creating web sites to deliver faster results
- Advanced visual diagnostics speed debugging
- It offers team Foundation Server integration
- Rich standards-based web design & technologies
- It provides precision layout control and supports a broad range of technologies.
- It helps to speeds up cross-browser debugging with advanced visual diagnostics

13. HTML5 Builder

HTML5 Builder is a software solution for building the web and mobile apps. It can develop an app using a single HTML5, CSS3, JavaScript and PHP codebase. It helps to target multiple mobile operating systems, devices and Web browsers.

Features:

- It is the fastest way to develop cross-platform Apps with flexible Cloud services
- Increased speed of development with a single visual framework
- Brings Designers and Developers in a Collaborative Workflow
- Create Enterprise or ISV web and mobile apps
- Create location-based browser and mobile applications using geolocation components in HTML5 Builder

14. Visual Online

Visual Studio Online is a collection of services. It is fast and easy to plan, build and ship software across a variety of platforms. These software development tools allow the organization to create the perfect development environment.

Features:

- Track and manage all ideas on kanban or scrum boards with agile tools
- Improve code quality and catch issues early
- Build, manage, secure and share software components
- Automate and simplify Azure deployments
- Tools for manual, performance and automated testing
- It offers a centralized version control system with free private repositories.

Cloud Tools

15. Kwatee

Kwatee Agile Deployment is a software development tool. It automates applications or micro services to any number of servers. It fully automates deployments of text and binary files from any number of target servers.

Features:

- It allows managing environment-specific configuration parameters for your application
- It allows generating command-line installers to be used for deployments on environments
- This software development tool eliminates the need for installation & configuration. It also takes out many operational risks in the software development process
- It is a friendly web interface lets configure deployments efficiently and painlessly
- It supports a broad range operating systems including Linux, Windows, Mac OS X, Solaris, etc.

16. Azure

Microsoft Azure is widely used by developers to build, deploy and manage web applications.

Features:

- It supports wide range of operating systems, programming languages, frameworks, and devices
- Allows to build apps quickly and easily
- It easily detect and mitigate threats
- Rely on the most trusted cloud
- Allows to manage app proactively
- Helps to deliver mobile apps seamlessly

Data Science

17. Data studio

Dataiku DSS is a collaborative data science software platform. It is used by data scientists, data analysts, and engineers to explore, prototype, build and deliver their data products.

Features:

- Profile the data visually at every stage of the analysis

- Prepare, enrich, blend, and clean data using more than 80+ built-in functions
- Bundle whole workflow as a single deployable package for real-time predictions
- Build & optimize models in Python or R and integrate any external ML library through code APIs

Source Control

18. Github

GitHub allows developers to review code, manage projects, and build software. It offers right tool for different development jobs.

Features:

- Coordinate easily, stay aligned, and get done with GitHub's project management tools
- Easy documentation alongside quality coding
- Allows all code in a single place
- Developers can host their documentation directly from repositories

19. BitBucket

Bitbucket is a version control tool. It facilitates easy collaboration amongst software development team. It integrates very well with JIRA, a famous project and issue-managing app.

Features:

- Branch permissions access to ensure that the only right people can make changes to the code
- Helps development team to focus on a goal, product or process by organizing repositories into projects
- It can display build results from CI system
- Integrate into existing workflow to streamline software development process

20. Cloudforge

CloudForge is a software-as-a-service product for application development. It Integrates and manages various development tools.

Features:

- Integrate and manage various development tools
- Elastically scale development teams, projects, and processes
- Deploy code to public and private clouds
- Deploy source-code with a single click using various protocols including FTP, SCP, SSH, and Rsync to run-time environment
- CloudForge Publisher allows deployment to multiple servers in parallel

Prototyping

21. Axure

Axure provides the capability to produce wireframes, prototypes, and create documentation. This tool is used by business analysts, product managers, and IT consultants around the world.

Features:

- Axure RP generates prototype in HTML and provides link for sharing
- It allows multiple people to work on the same file at the same time
- It can runs on Microsoft IIS with a MySQL or Microsoft SQL Server database
- It helps to create and maintain widget libraries

DevOps

22. Codenvy

Codenvy automates applications or micro services to any number of servers. It fully automates deployments of text and binary files from any number of target servers.

Features:

- It allows managing environment-specific configuration parameters for your application
- It allows generating command-line installers to be used for deployments on any environment
- This software development tool eliminates the need for installation & configuration.
- It takes out many operational risks in the software development process
- The friendly web interface lets configure deployments efficiently and painlessly
- It supports a broad range operating systems including Linux, Windows, Mac OS X, Solaris, etc.

Notifications

23. SendBird

Sendbird is used as a messaging and Chat API for Mobile Apps and Websites. It offers scalability for a massive audience. It also prevents spam flooding of chat rooms.

Features:

- Read and track the status of the messages sent to users
- Integrate bots to assist with customer support and product recommendations
- Offers Push Notifications &Callbacks
- Read Receipt & Delivery Status
- Automatically split or merge chat rooms on the audience volume to offer continues engagement

UML

24. Enterprise Architect

Enterprise Architect is a requirement management tool. It integrates seamlessly with other development tools by creating requirements in the model.

Features:

- Allows to build robust and maintainable systems
- It loads extremely large models in seconds
- Collaborate effectively globally
- Offers complete traceability
- Improve business outcomes
- Model and manage complex Data effectively
- Supports Single click HTML and document generation
- Code execution to visual diagrams

SELECTING LANGUAGE AND TOOLS

Since, many programming languages and tools provide the same specialties in their form. Most of them can be used as substitute of other. So, it's a difficult task to select which tool and language is to better to use in your software production process. The first choice can be done on the basic of easiness to the team members who are working on the project. Sometimes the choice depends on the demand of the stakeholder of the system like the client. Similarly we can also make a choice according to the availability of the tool and ease of handling them. The other factor of selection may lead by the nature of the system that we need to develop.

Hence, before selecting the language and tools better to study the client's requirements, nature of the system to be developed, ease and proficiency of the team members involved in the development process and availability of the tools is also needed to be checked. By incorporating these factors we can select better tools and languages that we need for the system development.

Good Program Practice

Good Programming Practice is important within the software industries as an increased need for efficiency means that code that is clear, easy to maintain and efficient is more important than ever. Efficient code and best practices should not conflict with one another. It is essential to have various guidelines to govern and regulate code on clarity, efficiency, re-usability, adaptability and robustness.

Good Programming Practices:

- Ensure the clarity of the code and facilitate code review;
- Save time in case of maintenance, and ease the transfer of code among programmers or companies;

- Minimize the need for code maintenance by robust programming;
- Minimize the development effort by development and re-use of standard code and by use of dynamic (easily adaptable) code;
- Minimize the resources needed at execution time (improve the efficiency of the code);
- Reduce the risk of logical errors.
- Meet regulatory requirements regarding validation and 21CFRPart11 compliance

The figure below illustrates a \taxonomy that assists developers to create good software in categories. The categories cover different aspects of writing any software and they are based on software engineering, industry, and coding experiences. By following the tasks illustrated in figure below we can develop well-managed software. The tasks are divided into three different categories as: Before writing Software, While writing the Software and after writing Software. These categories hold the task to be done accordingly.

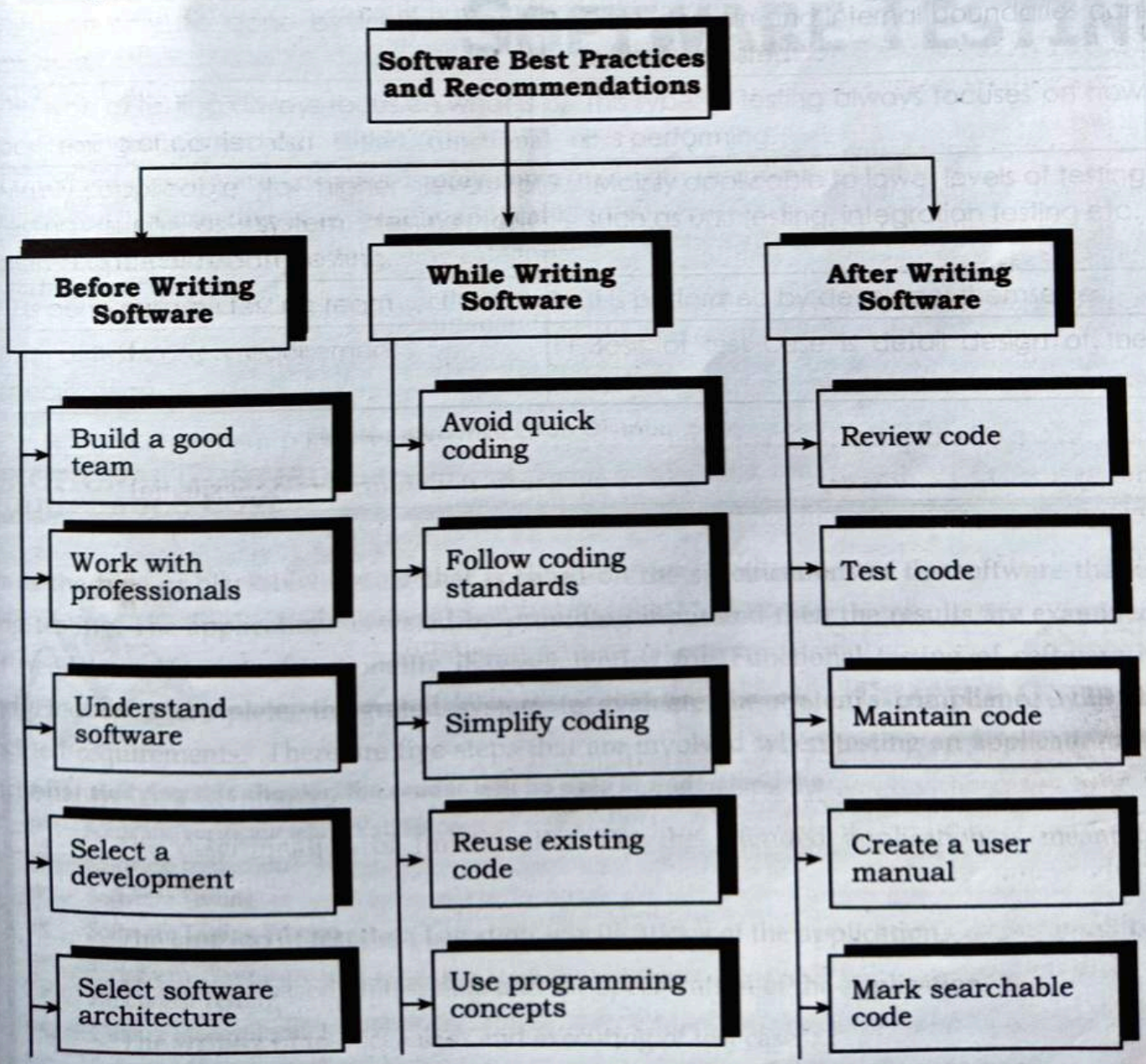


Fig: Good Programming Practices Flowchart

EXERCISE

1. What is programming language? Explain various software development tools.
2. Explain good programming practices used in software development.

❑❑❑