PROGRAM DEVELOPMENT AND PROGRAMMING LANGUAGES

Application of Information and Communication Technologies

Dr. Muhammad Abdullah

Faculty of Computing and Information Technology (FCIT)
University of the Punjab, Lahore, Pakistan.

Learning Objectives

1. Understand the differences between structured programming, object-oriented programming (OOP), aspect-oriented programming (AOP), and adaptive software development.
2. Identify and describe the activities involved in the program development life cycle (PDLC).
3. Understand what constitutes good program design and list several tools that can be used by computer professionals when designing a program.

Learning Objectives

4. Explain the three basic control structures and how they can be used to control program flow during execution.
5. Discuss some of the activities involved with debugging a program and otherwise ensuring it is designed and written properly.
6. List some tools that can be used to speed up or otherwise facilitate the program development process.
7. Describe several programming languages in use today and explain their key features.

Overview

- This chapter covers:
  - The most common approaches to program design and development
  - The phases of the program development life cycle (PDLC)
  - Tools that can be used to design and develop a program
  - Good program design techniques and types of program errors
  - Common programming languages

Approaches to Program Design and Development

- **Procedural programming:** An approach to program design in which a program is separated into small modules that are called by the main program or another module when needed
  - Uses procedures (modules, subprograms): Smaller sections of code that perform specific tasks
  - Allows each procedure to be performed as many times as needed; multiple copies of code not needed
  - Prior to procedural programming, programs were one large set of instructions (used GOTO statements)
  - **Structured programming**: Goes even further, breaking the program into small modules (Top-down design)

Approaches to Program Design and Development

- Variables: Named memory locations that are defined for a program
  - Used to store the current value of data items used in the program

Approaches to Program Design and Development

- **Object-oriented programming (OOP):** Programs consist of a collection of objects that contain data and methods to be used with that data
  - Class: Group of objects that share some common properties
  - Instance: An individual object in a class
  - Attributes: Data about the state of an object
  - Methods: Perform actions on an object
  - Objects can perform nontraditional actions and be easily used by more than one program

Approaches to Program Design and Development

- **Aspect-oriented programming** (AOP): Separates functions so program components can be developed and modified individually from one another
  - The components can be easily reused with separate nonrelated objects
- **Adaptive software development**: Designed to make program development faster and more efficient and focus on adapting the program as it is being written
  - Iterative and/or incremental
  - Includes RAD (rapid application development) and extreme programming (XP)
  - Agile software development: Focuses on building small functional program pieces during the project

The Program Development Life Cycle (PDLC)

- **Program development:** The process of creating application programs

- Program development life cycle (PDLC): The process containing the five phases of program development

The Program Development Life Cycle (PDLC)

- **Problem analysis**: The problem is considered and the program specifications are developed
  - Specifications developed during the PDLC are reviewed by the systems analyst and the programmer (the person who will code the program)
  - Goal: To understand the functions the software must perform
  - Documentation: Includes program specifications (what it does, timetable, programming language to be used, etc)

The Program Development Life Cycle (PDLC)

- **Program design**: The program specifications are expanded into a complete design of the new program
  - Good program design is extremely important
  - Program design tools
    - Structure charts: Depict the overall organization of a program
    - **Flowcharts:** Show graphically step-by-step how a computer program will process data
      - Use special symbols and relational operators
      - Can be drawn by hand or with flowcharting software

Flowcharts

Flowcharts

The Program Development Life Cycle (PDLC)

- **Pseudocode**: Uses English-like statements to outline the logic of a program
- **Unified Modeling Language (UML) Models:** Set of standard notations for creating business models
  - Widely used in object-oriented programs
  - Includes class diagrams, use case diagrams, etc.

Unified Modeling Language (UML) Models

The Program Development Life Cycle (PDLC)

- **Control structure**: A pattern for controlling the flow of logic in a computer program, module, or method
  - Sequence control structure: Series of statements that follow one another
  - Selection control structure: Multiple paths, direction depends on result of test
    - If-then-else
    - Case (avoids nested if-then-else statements)
  - Repetition control structure: Repeat series of steps
    - Do-while
    - Do-until

Control Structures

Control Structures

The Program Development Life Cycle (PDLC)

- Good program design:
  - Is essential
  - Saves time
- Good program design principles:
  - Be specific
    - All things the program must do need to be specified
  - One-entry-point/one-exit-point rule
  - No infinite loops or other logic errors
    - Infinite loop: Series of steps that repeat forever
- Design should be tested to ensure logic is correct
  - Desk check; tracing tables
- Documentation: Includes design specifications

Good Program Design

Program Design Testing

Program Design Testing

The Program Development Life Cycle (PDLC)

- **Program coding**: The program code is written using a programming language.
  - When choosing a programming language, consider:
    - Suitability to the application
    - Integration with other programs
    - Standards for the company
    - Programmer availability
    - Portability if being run on multiple platforms
    - Development speed
  - Coding creates source code

Coding Standards

- Coding standards: Rules designed to standardize programming
  - Makes programs more readable and easier to maintain
  - Includes the proper use of comments to:
    - Identify the programmer and last modification date
    - Explain variables used in the program
    - Identify the main parts of the program
- Reusable code: Pretested, error-free code segments that can be used over and over again with minor modifications
  - Can greatly reduce development time
- Documentation: Includes documented source code

Comments

The Program Development Life Cycle (PDLC)

- Program debugging and testing: The process of ensuring a program is free of errors (bugs) and works as it is supposed to
  - Before they can be debugged, coded programs need to be translated into executable code
    - Source code: Coded program before it is compiled
    - Object code: Machine language version of a program
    - Language translator: Program that converts source code to machine language

The Program Development Life Cycle (PDLC)

- Types of language translators:
  - Compilers: Language translator that converts an entire program into machine language before executing it
  - Interpreters: Translates one line of code at one time
  - Assemblers: Convert assembly language programs into machine language

The Program Development Life Cycle (PDLC)

- Preliminary debugging: Finds initial errors
  - Compiler errors: Program doesn't run
    - Typically syntax errors: When the programmer has not followed the rules of the programming language
  - Run time error: Error that occurs when the program is running
  - Logic errors: Program will run but produces incorrect results
    - Dummy print statements can help locate logic errors and other run time errors

Preliminary Debugging

Preliminary Debugging

The Program Development Life Cycle (PDLC)

- Testing: Occurs after the program appears to be correct to find any additional errors
  - Should use good test data
  - Tests conditions that will occur when the program is implemented
  - Should check for coding omissions (product quantity allowed to be < 0, etc.)
  - Alpha test (inside organization)
  - Beta test (outside testers)
- Documentation: Completed program package (user's manual, description of software commands, troubleshooting guide to help with difficulties, etc.)

The Program Development Life Cycle (PDLC)

- Program implementation and maintenance: Installing and maintaining the program
  - Once the system containing the program is up and running, the implementation process is complete
  - Program maintenance: Process of updating software so it continues to be useful
    - Very costly
  - Documentation: Amended program package

Quick Quiz

1. Which approach to programming uses the concept of inheritance?
   a. Procedural
   b. Object-oriented
   c. Aspect-oriented
2. True or False: An infinite loop is an example of a logic error.
3. A(n)_____ is a program design tool that shows graphically step-by-step the actions a computer program will take.

Answers:

1) b; 2) True; 3) flowchart

Tools for Facilitating Program Development

- Application Lifecycle Management (ALM): Creating and managing an application during its entire lifecycle, from design through retirement
  - Tools include:
    - Requirements management: Keeping track of and managing the program requirements as they are defined and then modified
    - Configuration management: Keeping track of the progress of a program development project
    - Issue tracking: Recording issues such as bugs or other problems that arise during development or after the system is in place

Tools for Facilitating Program Development

Tools for Facilitating Program Development

- Application generator: Software program that helps programmers develop software
  - Macro recorders: Record and play back a series of keystrokes
  - Report and form generators: Tools that enable individuals to prepare reports and forms quickly

Tools for Facilitating Program Development

- Device software development tools: Assist with developing embedded software to be used on devices, such as cars, ATM machines, consumer devices, etc
- Software development kits (SDKs): Designed for a particular platform; enables programmers to develop applications more quickly and easily
  - Released by hardware or software companies
  - e.g. iPhone SDK
- Application Program Interfaces (APIs): Help applications interface with a particular operating system
  - Often used in conjunction with Web sites

Tools for Facilitating Program Development

- Rich Internet Application (RIA): Web-based applications that work like installed software programs
  - Desktop RIA can access local files and used without an Internet connection
  - Web-based RIAs are common
  - Tools to develop RIAs
    - Adobe AIR

Quick Quiz

1. Which of the following is not an Application Lifecycle Management (ALM) tool?

a. Requirements definition software

b. Code generator

c. Application program interface (API)

2. True or False: A software development kit (SDK) is designed for a particular platform and allows programmers to develop applications quickly for that platform.

3. A(n) _____ is often used to create the forms or input screens used to input data into a program or database.

Answers:

1) c; 2) True; 3) form generator

Programming Languages

- Programming language: A set of rules, words, symbols, and codes used to write computer programs
  - To write a program, you need appropriate software for the programming language you will be using
- Categories of programming languages
  - **Low-level languages**: Difficult to code in; machine dependent
    - Machine language: 1s and 0s
    - Assembly language: Includes some names and other symbols to replace some of the 1s and 0s in machine language

Programming Languages

Programming Languages

- **High-level languages:** Closer to natural languages
  - Machine independent
  - Includes 3GLs (FORTRAN, BASIC, COBOL,C, etc.) and object-oriented languages (Visual Basic, C#, Python, Java, etc.)
  - Visual or graphical languages: Use graphical interface to create programs
- **Fourth-generation languages (4GLs):** Even closer to natural languages and easier to work with than high-level
  - Declarative rather than procedural
  - Includes structured query language (SQL) used with databases

Common Programming Languages

- FORTRAN: High-level programming language used for mathematical, scientific, and engineering applications
  - Efficient for math, engineering and scientific applications
  - Still used today for high-performance computing tasks (weather forecast)

Common Programming Languages

- COBOL: Designed for business transaction processing
  - Makes extensive use of modules and submodules
  - Being phased out in many organizations
  - Evolving (COBOL.NET)

Common Programming Languages

- Pascal: Created as a teaching tool to encourage structured programming
  - Contains a variety of control structures used to manipulate modules systematically

Common Programming Languages

- BASIC: Easy-to-learn, high-level programming language that was developed to be used by beginning programmers
  - Visual Basic: Object-oriented version of BASIC; uses a visual environment

Common Programming Languages

- C: Designed for system programming
- C++: Object-oriented versions of C
- C#: Used for Web applications
- Objective-C: For iPhone and other Apple applications

Common Programming Languages

- Java: High-level, object-oriented programming language frequently used for Web-based applications
  - Java programs are compiled into bytecode
  - Can run on any computer that includes Java Virtual Machine (Java VM)
  - Can be used to write Java applets
    - Scroll text on Web page, games, calculators, etc
  - Is one of the most popular programming languages today

Common Programming Languages

Common Programming Languages

- Python: Open-source, dynamic, object-oriented language that can be used to develop a variety of applications
  - Gaming, scientific, database, and Web applications
  - Only recently gaining a following

Common Programming Languages

- Ruby: Open-source, object-oriented language that can be used to create general-purpose or Web applications
  - Uses a syntax that is fairly easy to read and write, allowing programmers to create database-driven Web applications easily and quickly

Quick Quiz

1. An example of a high-level programming language is _____.
   a. Pascal
   b. Assembly language
   c. Machine language
2. True or False: Visual Basic is an object-oriented version of COBOL.
3. Java applets are small programs written in the _____ programming language.

Answers:

1) a; 2) False; 3) Java

Summary

- Approaches to Program Design and Development
- The Program Development Life Cycle (PDLC)
- Tools for Facilitating Program Development
- Programming Languages