A Guide to Programming Languages: One Size Doesn’t Fit All

- **Web-based languages**
  - Not considered programming languages
  - Tell the browser how to display text and objects
  - Language types
    - Markup
    - Scripting
A Guide to Programming Languages: One Size Doesn’t Fit All

- **Markup languages**
  - Composed of a set of codes, or *elements*, that uses tags to define how text and objects display
    - **Tags**—markers that usually come in pairs
    - **Content**—text that displays, lies between the opening and closing tags
A Guide to Programming Languages: One Size Doesn’t Fit All

• **Markup languages**
  o **HTML (Hypertext Markup Language)**
    • Used by Web programmers
    • Supports links to other documents, graphics, and audio and video files.
  o **XML (Extensible Markup Language)**
    • Enables programmers to capture specific types of data by creating their own elements
    • Not a replacement for HTML
  o **XHTML (Extensible Hypertext Markup Language)**
    • Newer version of HTML that uses XML to produce Web pages that are easily accessible by newer portable devices.
A Guide to Programming Languages: One Size Doesn’t Fit All

• **Scripting languages**
  o Create **scripts**—programs that control Web page actions or responses
    • VBScript
    • ActiveX controls
    • JavaScript
    • ECMAScript
    • AJAX
    • JSON
      o Used for **serialization** and transmitting structured data over a network
  • PHP
The Program Development Life Cycle

• Program development life cycle (PDLC)
  o Organized plan for managing the development of software
  o Consists of six phases, from problem definition through program implementation and maintenance
The Program Development Life Cycle

- **PHASE 1**: Defining the problem
- **PHASE 2**: Designing the program
- **PHASE 3**: Coding the program
- **PHASE 4**: Testing and debugging the program
- **PHASE 5**: Documenting the program
- **PHASE 6**: Implementing and maintaining the program
The Program Development Life Cycle

• **Phase 1: Defining the problem**
  
  o Define the problem the program will solve
  
  o Define the **program specifications**, including decisions regarding data input, required processing, output, and the user interface
The Program Development Life Cycle

• **Phase 2: Designing the program**
  
  o **Program design**—identifies components of the program

  • **Top-down program design**—breaks program into small, manageable, highly focused routines
    
    o Procedures, functions, or subroutines
  
  • **Structured design** uses **control structures**—logical elements assembled in blocks of code that determine how subroutines will be programmed
The Program Development Life Cycle

• **Phase 2: Designing the program (con’t.)**
  
  o Basic control structures categories
    
    • **Sequence control structure**—code performed in line-by-line order
    
    • **Selection control structure**—also called a conditional or branch structure, this is a portion of code that leads to a block of code based on conditions being met
    
    • **Case control structure**—portion of code that branches to extensive conditional coding
    
    • **Repetition control structure**—also known as looping or iteration, this is a portion of code that repeats
The Program Development Life Cycle

• **Phase 2: Designing the program (con’t.)**
  
  o **Algorithm**
    • Combination of control structures
    • Step-by-step description of how to arrive at a solution
  
  o **Nesting**
    • Process of embedding control structures within one another
The Program Development Life Cycle

- Control module
  - Print report heading
  - Process each separation
  - Print total wages
    - Calculate base salary
    - Calculate commission
      - Calculate total wages = base salary + commission
The Program Development Life Cycle

• **Phase 2: Designing the program (con’t.)**
  
  o Program design tools
  
    • **Structured charts**—also called **hierarchy charts**, show top-down design of programs
    
    • **Flowcharts**—use diagrams to show the logic of a program
    
    • **Unified Modeling Language (UML)**—variation of flowcharting used to illustrate and document object-oriented systems during development
    
    • **Pseudocode**—uses a stylized form of writing to describe logic
The Program Development Life Cycle

Start commission program.

Calculate base salary.

If sales exceed 100,000:

- Yes: Calculate commission.
- No: Commission = 0.

Total salary = base salary + commission.

Print total salary.

End commission program.
The Program Development Life Cycle

- **Phase 3: Coding the program**
  - Programmers convert algorithms into programming code
  - **Syntax errors**
    - Mistakes in the construction of the programming commands
    - Must be corrected for the program to run appropriately
The Program Development Life Cycle

• **Phase 4: Testing and debugging the program**
  
  o All errors, not just syntax errors, must be removed
  
  o **Logic errors**
    
    • Relate to problems in the solution’s design
    • Cause incorrect output
    • Program still runs despite logic errors
  
  o Syntax errors and logic errors—**bugs**
  
  o **Debugging**—process of eliminating errors
The Program
Development Life Cycle

• **Phase 5: Documenting the program**
  
  o **Documentation** includes:
    
    • Overview of program functionality
    • Tutorials
    • Thorough explanation of main features
    • Reference documentation of program commands
    • Description of error messages
    • Program design work, including structure charts, pseudocode, and flowcharts
The Program Development Life Cycle

- **Phase 6: Implementing and maintaining the program**
  - Test the program
    - Have users work with the software
    - Correct errors
  - **Program maintenance**
    - Fix program errors discovered by users
    - Conduct periodic evaluations on a regular basis
    - Make modifications as needed to update the program or add features