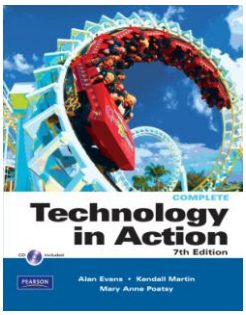


Technology in Action

Chapter 10

Behind the Scenes: Building Applications

Chapter Topics

- System development life cycle
- Life cycle of a program
- Problem statement
- Algorithms
- Moving from algorithm to code

Chapter Topics (cont.)

- Moving from code to machine language
- Testing programs
- Completing a program
- Selecting the right programming language
- Most popular programming languages

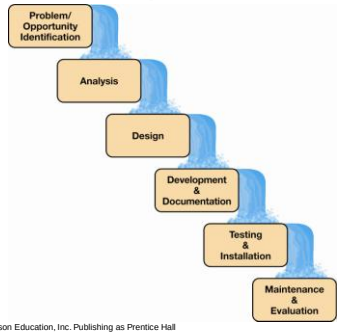
Information Systems

- System
  - A collection of pieces working together to achieve a common goal
- An information system includes
  - Data
  - People
  - Procedures
  - Hardware and software

Reasons for Software Programming

- Some types of tasks are candidates for automation as a software program
  - Routine
  - Repetitive
  - Work with electronic data
  - Follow a series of clear steps
- A new software program can be created when existing programs do not suffice.

System Development Life Cycle



Problem and Opportunity Identification

- The existing system is evaluated
  - Problems are defined
  - New proposals are reviewed
  - Decisions are made to proceed with the projects
  - The process is documented
  - Relevant problems and opportunities are defined

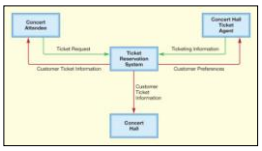
Analysis

- A program specification (goals and objectives of the project) is developed
- A feasibility assessment is performed
- User requirements are defined
- Analysts recommend a plan of action

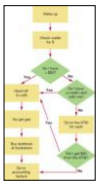
Design

- A detailed plan for programmers is developed
- Flowcharts and data-flow diagrams are used for the current and proposed system

Data-flow diagram



Flowchart



Development and Documentation

- Actual programming takes place
- First phase of the program development life cycle (PDLC)
- Development is documented
- User documentation is created

Testing and Installation

- Program is tested for proper operation
- Program is installed for use
- Testing and results are documented

Maintenance and Evaluation

- Performance of the system is monitored
- Corrections and modifications to the program are made
- Maintenance procedures and results are documented

Ethics in IT

- The Association of Computing Machinery (ACM) and the Institute of Electrical and Electronic Engineers (IEEE) have established eight principles for ethical software engineering practices:
  1. Public
  2. Client and Employer
  3. Product
  4. Judgment

Ethics in IT

- Ethical software engineering practices (cont.):
  5. Management
  6. Profession
  7. Colleagues
  8. Self

Joint Application Development (JAD)

- Helps designers adapt to changes in program specifications
- Includes customer involvement
- No communication delays
- Also referred to as:
  - Accelerated design
  - Facilitated team technique

The Life Cycle of a Program

- Programming is the process of translating a task into a series of commands a computer will use to perform that task
- Programming involves
  - Identifying the parts of a task the computer can perform
  - Describing tasks in a specific and complete manner
  - Translating the tasks into a language understood by the computer's CPU

Program Development Life Cycle

Step 1: Describing the Problem

- The problem statement:
  - Is the starting point of programming
  - Describes tasks the program is to accomplish
  - Describes how the program will execute the tasks
  - Is created through interaction between the programmer and the user
  - Includes error handling, a testing plan, and output values

Parking Garage Example

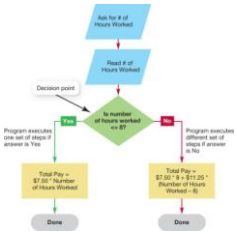
Program Goal:	To compute the total pay for a fixed number of hours worked at a parking garage.		
Inputs:	Number of Hours Worked..... a positive number		
Outputs:	Total Pay Earned ..... a positive number		
Process:	The Total Pay Earned is computed as $7.50 per hour for the first eight hours worked each day. Any hours worked beyond the first eight are billed at $11.25 per hour.		
Error Handling:	The input Number of Hours Worked must be a positive real number. If it is a negative number or other non-acceptable character, the program will force the user to re-enter the information.		
Testing Plan:	**INPUT**	**OUTPUT**	**NOTES**
	8	8*7.50	Testing positive input
	3	3*7.50	Testing positive input
	12	8*7.50 + 4*11.25	Testing overtime input
	−6	Error message/ask user to reenter value	Handling error

Step 2: Developing an Algorithm

- Algorithm development
  - A set of specific, sequential steps that describe what the program must do
  - Complex algorithms include decision points
    - Binary (yes/no)
    - Loop (repeating actions)
  - Visual tools used to track algorithm and decision points

Flowchart and Pseudocode

Flowchart



Pseudocode

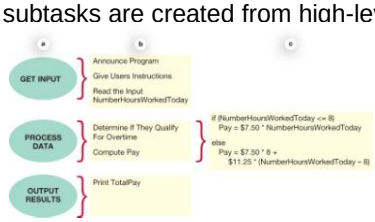
Bold terms show actions that are common in programming, such as reading data, making decisions, printing, and so on.

1. **Ask the user** how many <u>hours they worked</u> today
2. If the number of hours worked < = 8,
   compute <u>total pay</u> without overtime
   otherwise,
   compute <u>total pay</u> with overtime pay
3. **Print** <u>total pay</u>

<u>Underlined</u> words are information items that appear repeatedly in the algorithm.

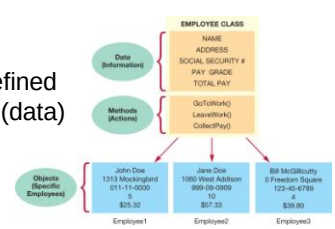
Top-Down Design

- Problem is divided into a series of high-level tasks
- Detailed subtasks are created from high-level tasks



Object-Oriented Analysis

- Classes (categories of inputs) are identified
- Classes are defined by information (data) and actions (methods or behaviors)
- Reusability is key



Step 3: Coding

- Coding is translating an algorithm into a programming language
- Generations of programming languages
  - 1GL: Machine
  - 2GL: Assembly
  - 3GL: FORTRAN, BASIC, C, Java
  - 4GL: SQL
  - 5GL: PROLOG

Compilation

- Compilation is the process of converting code into machine language
- The compiler reads the source code and translates it into machine language
- After compilation, programmers have an executable program

Interpreter

- Some programming languages do not have a compiler but use an interpreter instead
  - The interpreter translates source code into a line-by-line intermediate form
  - Each line is executed before the next line is compiled
  - Programmers do not have to wait for the entire program to be recompiled each time they make a change
  - Programmers can immediately see the results of changes as they are making them

Coding Tools: Integrated Development Environments

- Editor: Special tool that helps programmers as they enter the code
- Debugging: Removal of errors in code
  - Syntax error: Mistake in use of the language
  - Logic error (runtime error): Mistake in the algorithm

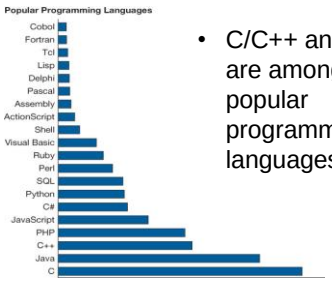
Step 4: Debugging

- Running a program to find errors is known as debugging
- Sample inputs are used to determine runtime (logic) errors
- Debugger: Tool that helps programmers locate runtime errors

Step 5: Finishing the Project

- Users test the program (internal testing)
- Beta version released
  - Information collected about errors before final revision
- Software updates (service packs)
  - Problems found after commercial release
- Documentation created
  - User manuals
  - User training

Popularity of Programming Languages



- C/C++ and Java are among the most popular programming languages.

Programming Languages

- Selecting the right language
  - Space available
  - Speed required
  - Organizational resources available
  - Type of target application

Windows Applications: Visual Basic 2008

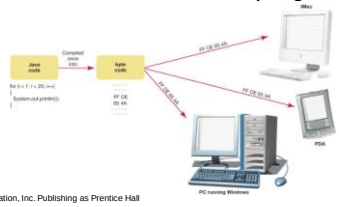
- Used to build Windows applications
- Object-oriented language
- Visual Basic 2008 is the current version

C and C++

- C
  - Developed for system programmers
  - Combines high- and low-level programming features
  - Modern operating systems are written in C
- C++
  - Uses the same features as C
  - Includes object-oriented design

Java

- Object-oriented features
- Large set of existing classes
- Architecture neutral
- Java applets: Small Java-based programs



Web Applications

- HTML/XHTML
  - HyperText Markup Language/eXtensible HyperText Markup Language
  - Not a true programming language
  - Uses special symbols (tags) to control how Web pages are viewed
- eXtensible Markup Language (XML)
  - Enables computers to efficiently transfer information between Web sites

Web Applications

- Scripting languages: Limited to performing a specific set of specialized tasks
  - JavaScript: Used to make Web pages more visually appealing and interactive
  - VBScript: Subset of VB used to add interactivity to Web pages
  - PHP: Another scripting language gaining in popularity
- Dynamic decision making
  - Web page can display content based on user choices

HTML/XHTML Editors

- Popular tools for creating Web pages
  - Adobe Dreamweaver
  - Microsoft Expression Web
- No programming is required.

Web Applications

- Active Server Pages (ASP), Java Server Pages (JSP), and PHP
  - Add interactivity to Web pages
  - Translate user information into a request for more information from a company's computer
- XML
  - Enables designers to define their own data-based tags

Adobe Flash and Microsoft SilverLight

- Flash
  - Used to develop Web-based multimedia
  - Includes its own scripting language, ActionScript
- SilverLight
  - Supports development of multimedia and interactive Web applications

AJAX

- AJAX (Asynchronous JavaScript And XML)
  - Uses a combination of existing technologies like JavaScript, CSS, and XML
  - Allows for information updates without a page refresh
  - Allows for a more responsive user experience

The Next Great Language

- Large projects may take 30 minutes to compile
- Interpreted languages might become more important
  - Python
  - Ruby
  - Smalltalk

Blender

- Video game development tool
- Open source
- Built-in game engine
- Built-in physics engine
- Uses logic bricks to simplify programming

Chapter 10 Summary Questions

- What is a system development life cycle, and what are the phases in the cycle?

Chapter 10 Summary Questions

- What is the life cycle of a program?

Chapter 10 Summary Questions

- What role does a problem statement play in programming?

Chapter 10 Summary Questions

- How do programmers create algorithms?

Chapter 10 Summary Questions

- How do programmers move from algorithm to code, and what categories of language might they code in?

Chapter 10 Summary Questions

- How does a programmer move from code in a programming language to the 1s and 0s the CPU can understand?

 49

Chapter 10 Summary Questions

- How is a program tested?

 50

Chapter 10 Summary Questions

- What steps are involved in completing the program?

 51

Chapter 10 Summary Questions

- How do programmers select the right programming language for a specific task?

 52

Chapter 10 Summary Questions

- What are the most popular Windows and Web applications?

 53