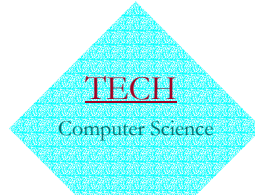


CH07: Writing the Programs

- Does not teach you how to program, but point out some software engineering practices that you should keep in mind as you write your code.
- * Programming Standards and Procedures
- * Programming Guidelines
- * Documentation



Programming Standards and Procedures

- Most software is developed by teams
- Standards for you
 - force you to organize!
- Standards for others to understand
 - (1) what you have written
 - (2) why you have written it
 - (3) how it fits in with other work

Standards format for comments, e.g.

- /* Statement of function:
- * Component name:
- * Programmer:
- * Version:
- * Procedure Invocation:
- * Input Parameters:
- * Output Parameters:
- */

Matching Design with Implementation

- Direct correspondence between the program design and program code
- Configuration Management for tracing the correspondence

Programming Guidelines

- Major aspects of programming:
 - control structures
 - algorithms
 - data structures

Control Structures

- (1) preserve the control structure suggested by architecture and design
- (2) should not jump wildly through the code
- (3) modularity
- (4) generality is a virtue (do not over do it)
- (5) dependence among components must be visible

Algorithms

- Choose wisely!
- Algorithms effects execution time
- “do not sacrifice clarity and correctness for speed”

Data structures

- Keeping the program simple == give the data structures simple
- Using a data Structure to Determine a Program Structure
 - e.g. recursive data structure require recursive procedures

General Guidelines

- Localizing input and output
- Including pseudocode
 - outline what need to be done
 - a framework on which to construct the code
- Revising and Rewriting, not Patching
- Reuse

Documentation

- program documentation explain what the programs do and how they do it.
- Internal documentation is descriptive material written directly within the code: comments
- external documentation is descriptive material not within the code.

Comments (for yourself and others)

- Header comment block: overview of the component
- Other program comments, e.g.
 - // Increment i3
 - i3 = i3 + 1;
 - // Set counter to read next case
 - i3 = i3 + 1;
 - // Ideally
 - case_counter = case_counter + 1;

Meaningful Variable Names

- $weekWage = (hourRate * hours) + (0.5) * (hourRate) * (hours - 40);$
- // read this
- $z = (a * b) + (0.5) * (a) * (b - 40);$

Formatting to Enhance understanding

- indentation

Documenting data

- the way in which data are structured and used
- comment on the variable you declared

External Documents

- For people who may never look at the actual code
- describing interface:
 - **the inputs and the outputs**
 - **what arguments are passed**
 - **which arguments will be changed**
 - **what is returned**
- The processing
 - **how the inputs are transformed to outputs**