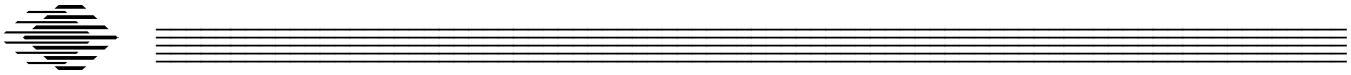
Assignment Kit for Coding/Counting Standard



PSP Fundamentals

The Software Engineering Institute (SEI)
is a federally funded research and development center
sponsored by the U.S. Department of Defense and
operated by Carnegie Mellon University.

PSP Fundamentals

Assignment Kit for the Coding/Counting Standard

Overview

Topics

This assignment kit covers the following topics.

Prerequisites

Prerequisites
- Read Chapter 4

Objectives

The objectives of the coding/counting standard are to
- establish a consistent set of coding practices
- provide criteria for judging the quality of the code that you produce
- facilitate size counting by ensuring you are consistent about what you put on each physical line

Coding/counting standard requirements

Produce, document, and submit a completed coding/counting standard that calls for quality coding practices.

For LOC counting, ensure that a separate physical source line is used for each logical line of code.

Submit the coding/counting standard with your program 2 assignment package.

Example coding/counting standard

Coding/counting standard example

The following pages contain a C++ coding/counting standard example.

Notes about the example:
- Since it is an example, tailor it to meet your personal needs.
- If you have an existing organizational standard, consider using it for the PSP exercises.

Continued on next page

C++ Coding/Counting Standard Example

Purpose	To guide implementation of C++ programs
Counting Standard	- Count each physical line as one LOC. - Do not count blank lines and comment-only lines. - Be consistent about what you put on each physical line.
Program Headers	Begin all programs with a descriptive header.
Header Format	/** / /* Program Assignment: the program number */ /* Name: your name */ /* Date: the date you started developing the program */ /* Description: a short description of the program and what it does */ /** /
Contents	Provide a summary of the contents
Contents Example	/** / /* Contents: */ /* Reuse instructions */ /* Modification instructions */ /* Compilation instructions */ /* Includes */ /* Class declarations: */ /* CData */ /* ASet */ /* Source code in c:/classes/CData.cpp: */ /* CData */ /* CData() */ /* Empty() */ /** /

(continued)

C++ Coding/Counting Standard Example (continued)

Reuse Instructions	- Describe how the program is used: declaration format, parameter values, types, and formats. - Provide warnings of illegal values, overflow conditions, or other conditions that could potentially result in improper operation.
Reuse Instruction Example	/*** / /* Reuse instructions */ /* int PrintLine(char *line_of_character) */ /* Purpose: to print string, 'line_of_character', on one print line */ /* Limitations: the line length must not exceed LINE_LENGTH */ /* Return 0 if printer not ready to print, else 1 */ /*** /
Identifiers	Use descriptive names for all variables, function names, constants, and other identifiers. Avoid abbreviations or single-letter variables.
Identifier Example	Int number_of_students; /* This is GOOD */ Float: x4, j, ftave; /* This is BAD */
Comments	- Document the code so the reader can understand its operation. - Comments should explain both the purpose and the behavior of the code. - Comment variable declarations to indicate their purpose.
Good Comment	If(record_count > limit) /* have all records been processed? */
Bad Comment	If(record_count > limit) /* check if record count exceeds limit */
Major Sections	Precede major program sections by a block comment that describes the processing done in the next section.
Example	/*** / /* The program section examines the contents of the array 'grades' and calcu- */ /* lates the average class grade. */ /*** /
Blank Spaces	- Write programs with sufficient spacing so they do not appear crowded. - Separate every program construct with at least one space.
Indenting	- Indent each brace level from the preceding level. - Open and close braces should be on lines by themselves and aligned.

Indenting Example	while (miss_distance > threshold)
	{
	success_code = move_robot (target _location);
	if (success_code == MOVE_FAILED)
	{
	printf("The robot move has failed.\n");
	}
	}

Capitalization	- Capitalize all defines. - Lowercase all other identifiers and reserved words. - To make them readable, user messages may use mixed case.
Capitalization Examples	#define DEFAULT-NUMBER-OF-STUDENTS 15 int class-size = DEFAULT-NUMBER-OF-STUDENTS;

Evaluation criteria and suggestions

Evaluation criteria	Your standard must be • complete • legible
Suggestions	Keep your standards simple and short. Do not hesitate to copy or build on the PSP materials.

Coding/Counting Standard Template

Purpose	To guide the development of programs
Counting Standard	- Count each physical line as one LOC. - Do not count blank lines and comment-only lines. - Be consistent about what you put on each physical line.
Program Headers	Begin all programs with a descriptive header.
Header Format	
Contents	Provide a summary of the contents.
Contents Example	
Reuse Instructions	- Describe how the program is used. Provide the declaration format, parameter values and types, and parameter limits. - Provide warnings of illegal values, overflow conditions, or other conditions that could potentially result in improper operation.
Reuse Example	
Identifiers	Use descriptive names for all variables, function names, constants, and other identifiers. Avoid abbreviations or single letter variables.
Identifier Example	

(continued)

Coding/Counting Standard Template (continued)

Comments	- Document the code so that the reader can understand its operation. - Comments should explain both the purpose and behavior of the code. - Comment variable declarations to indicate their purpose.
Good Comment	
Bad Comment	
Major Sections	Precede major program sections by a block comment that describes the processing that is done in the next section.
Example	
Blank Spaces	- Write programs with sufficient spacing so they do not appear crowded. - Separate every program construct with at least one space.
Indenting	- Indent every level of brace from the previous one. - Open and closing braces should be on lines by themselves and aligned with each other.
Indenting Example	
Capitalization	- Capitalized all defines. - Lowercase all other identifiers and reserved words. - Messages being output to the user can be mixed-case so as to make a clean user presentation.
Capitalization Example	

