Assignment Kit for Coding Standard

Personal Software Process for Engineers: Part I

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.

Personal Software Process for Engineers: Part I

Assignment Kit for the Coding Standard

Overview

Overview

This assignment kit covers the following topics.

Prerequisites

Prerequisites

- Read Chapter 4
- Complete Size Counting Standard

Objectives

The objectives of the coding 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 your programs are written so they can be readily counted
- for LOC counting, require that there be a separate physical line for each logical line of code

Coding standard requirements

Coding standard requirements	Produce, document, and submit a completed coding 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 standard with your program 2 assignment package.

Example coding Standard

Coding standard example

Pages 5 and 6 of this workbook contain an example C++ coding standard.

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

Example C++ Coding Standard

Purpose	To guide implementation of C++ programs
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 */ /**/
Listing Contents	Provide a summary of the listing contents
Contents Example	/**/ /* Listing Contents: */ /* Reuse instructions */ /* Modification instructions */ /* Compilation instructions */ /* Includes */ /* Class declarations: */ /* CData */ /* ASet */ /* Source code in c:/classes/CData.cpp: */ /* CData */ /* CData() */ /* Empty() */ /**/

(continued)

Example C++ Coding Standard (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 variable, 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 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 Standard Template

Purpose	Para guiar el desarrollo de programas en Java.
Program Headers	Comience todos los programas con un encabezado descriptivo.
Header Format	Asignación de programa: LOC Nombre: Miguel Romero Meza Fecha: 18/03/2021 Descripcion: contabilizador de lineas de trabajo de un programa
Listing Contents	Proporcione un resumen del contenido de la lista.
Contents Example	Clases declaradas: MAIN LOC Recursos de codigo en MAIN.java: main(String[] args) BuscarArchivos(File ruta,String fichero) Recursos de codigo en LOC.java: validarRuta(String ruta) reglas(String linea) ContadorLineasLOC(String ruta) identificarMetodos(String linea) datosMetodos(String linea) lienasmetodos(boolean identificador, String linea,int numero, int met)
Reuse Instructions	• Describe cómo se usa el programa. Proporcione el formato de declaración, los valores y tipos de los parámetros y los límites de los parámetros. • Brindar advertencias de valores ilegales, condiciones de desbordamiento u otras condiciones que podrían resultar en un funcionamiento incorrecto.
Reuse Example	• boolean identificarMetodos(String linea) • esta function recibe una linea e identifica que tenga las palabras reservadas de un metodo. • Limitacion: en caso de alguna declaracion de variable use estas mismas palabras sera contabilizada como metodo. • Regresa un valor booleano de haber identificado un metodo.
Identifiers	Utilice nombres descriptivos para todas las variables, nombres de funciones, constantes y otros identificadores. Evite las abreviaturas o las variables de una sola letra.
Identifier Example	Int NumLinea; /* Buen ejemplo */ int: met, ident; /* Mal ejemplo */

(continued)

Coding Standard Template (continued)

Comentarios	• Documente el código para que el lector pueda comprender su funcionamiento. • Los comentarios deben explicar tanto el propósito como el comportamiento del código. • Comentar declaraciones de variables para indicar su propósito.

Good Comment	//este metodo imprime el nombre y tipo de un metodo
	//recibe como parametron la linea donde se identidico a un metodo
	public void datosMetodos(String linea)

Bad Comment	public void datosMetodos(String linea) // regresa nombre y tipo de funcion
Major Sections	Preceder a las secciones principales del programa mediante un comentario en bloque que describe el procesamiento que se realiza en la siguiente sección

Example	//metodo que se encarga del conteo de lineas de lo archivos contenidos en la ruta que se le proporciona.
	//hace uso de otros metodos pertenecientes a la misma clase, que ayudan con la identificacion de algunos elementos
	public int ContadorLineasLOC(String ruta)

Blank Spaces	• Escriba programas con suficiente espacio para que no parezcan abarrotados. • Separe cada construcción de programa con al menos un espacio.
Indenting	• Identa cada nivel de llave del anterior y cada instruccion que pertenesca a un bloque. • Los aparatos de apertura y cierre deben estar alineados por sí mismos y alineados entre sí.

Indenting Example	public boolean reglas(String linea) {
	boolean comentario1 = linea.startsWith("/*");
	boolean comentario2 = linea.startsWith("//");
	boolean comentario3 = linea.startsWith("*/");
	boolean lineaVacia = (linea.length() == 0);
	
	if(!(comentario1 || comentario2 || comentario3 || lineaVacia)) {
	return true;
	}
	else {
	return false;
	}
	}

Capitalization	• Capitaliza todas las definiciones. • Ponga en minúscula todos los demás identificadores y palabras reservadas. • Los mensajes que se envían al usuario se pueden mezclar en mayúsculas y minúsculas para hacer una presentación clara del usuario.
Capitalization Example	System.out.println("Total de Items: "+contadorMetodos); System.out.println("Lineas Vacias: "+lineasvacias); System.out.println("Lineas de Codigo: "+contadorLineas);

