COP3503
Class Commenting Standards

Note: An example of the class commenting standards is provided on the next page for reference.

Required Comment Types

The following five commenting types are required:

Project Comment
The project comment should be located at the top of the class file containing the main() method, and should be in the Block style. The comment should include your name and the course number, as well as the project's number, title and due date. Finally, the comment should include a description of the program including its purpose, as well as any additional information the instructor might find interesting.

Class Comments
Class comments should be located above every class in the project, and should be in the JavaDoc style. The comments should include a description of the class' purpose, as well as any additional information, about the class, the instructor might find interesting. For projects with only one class, the description from the project comment can be copied for the class comment.

Method Comments
Method comments should be located above every method in the project, except the main() method, and should be in the JavaDoc style. The comments should include a description of the method's purpose, a description of each input parameter, as well as a description of what the method returns, if appropriate.

Section Comments
Section comments should be located above each logical section of code, and should be in the Line style. The comments should describe the purpose of their respective section of code.

Statement Comments
Statement comments should be located immediately above or to the right of a line of code, and should be in the Line style. By default, the comment should be located to the right of a line of code; however, if the comment runs "off-screen", it should be located immediately above the respective line of code. A statement comment should only be used when the purpose of a line of code may not be obvious to another programmer. As a general rule, less than 10% of a program's code should require statement comments.

Commenting Best Practices

1. Always align comments with the code they refer to.
2. Comments should serve to increase the readability and understanding of the code.
3. Comments should be short, but descriptive.
4. Comments should not be used for statements whose purpose is obviously clear.
5. Do not place blank lines between comments and their respective code.

Commenting Example

```
/*
 * Author:     Ima Java Programmer
 * Course:     COP3503
 * Project #: 1
 * Title   :   Rectangle A/P Calculator
 * Due Date:  1/1/2012
 *
 * Calculates the area of a rectangle.
 */

import java.util.Scanner;

/**
 * Calculates the area of a rectangle.
 */
public class RectangleClass {

    public static void main(String[] args) {
        // Variable declaration
        Scanner input = new Scanner(System.in);
        int length, width, area;

        // Get user input
        System.out.print("Enter the length of a rectangle: ");
        length = input.nextInt(); // Gets input from the keyboard
        System.out.print("Enter the width of a rectangle: ");
        width = input.nextInt();

        // Perform the calculation
        area = CalculateArea(length, width);

        // Display result to the user
        System.out.println("\nThe perimeter is " + perimeter);
    }

    /**
     * Calculates the area of a rectangle.
     * @param length The length of the rectangle.
     * @param width The width of the rectangle.
     * @return The area of the rectangle.
     */
    private static int CalculateArea(int length, int width) {
        int area  = length * width;

        return area;
    }
}
```

Project Comment

Class Comment

Statement Comment

Section Comment

Method Comment