Bayview Java Coding Standards

Naming

Classes	• CamelCase notation should be used. They begin with an uppercase letter. • Meaningful name should be chosen to reflect what the program does. • Avoid using acronyms or short forms – whole words are better form. • Avoid using underscores
Variables	• mixedCase notation for naming variables (first letter in first word should be lowercase, first letters in other words should be uppercase) – avoid using underscores • names should be meaningful and represent the data that they store and should not be too long • avoid using keywords as variable names
Constants	• names should be meaningful but not too long • ALL_CAPS and use of underscore to separate words is common practice.
Methods	• mixedCase notation is used for naming methods (first letter in the first word is lowercase, first letters in other words are uppercase) • names should be meaningful and represent the task that the method performs • verbs are good method names
Packages	• lowercase letters or mixedCase notation for naming packages

Blocks of Code and Curly Brackets {}, Bracket () Placement

** describe the placement of curly brackets both opening and closing

Curly Brackets:

- Opening bracket should be right after the class/method/loop/if statement and nothing else should be after it
- Closing bracket should on a line on its own – with a couple of exceptions: do-while loop and else if or else clauses

Bracket Placement:

- There should be a space before “(“ and after the “)” bracket
- Brackets should surround every conditional statement in a compound condition.

```
if ((number >= 0) && (number <= 10)) { // this is correct

if (number >= 0 && number <= 10) { // this is not proper style
```

Spacing and Indentation

** describe line spacing and indentation coding standards that have been discussed/observed in class

- Add a space around operators
- Add one blank line in between method
- Try to add blank lines to break up block of code where appropriate. Be careful too much white space makes it harder to read your code.

Declarations and Assignments

** describe best practices for variable declarations and assignments as discussed/observed in class

- One declaration or assignment per line – allows for comments to be added per declaration or as needed

Commenting/Documentation

** describe best practices for commenting as discussed/observed in class, include special symbols that are used for each type of comment

Class Header	• A block comment should be added at the beginning of each class including the name of the programmer, date, and description of the program
	`/**`
	` * Name: John Smith`
	` * Date: Apr 23, 2021`
	` * Description: This is an example of what a class`
	` *              header should look like.`
	` */`

Single line or inline	• Comments that are short and to the point and help explain what your program is doing or what a variable is used for `// this is an example of a inline comment`

Block comments	`/*`
	` * Block comments are used for comments`
	` * that span multiple lines.`
	` */`

Method headers	• Should be used to describe what the method is doing and define each parameter
	`/**`
	` * Calculates the square of a number.`
	` *`
	` * @param number - value to square`
	` * @return number squared`
	` */`
	`public static int square(int number) {`
	`  return number*number;`
	`}`