

CPT121 / COSC2135 Programming 1 (OUA)

Coding Style / Documentation Guide

Code formatting and indentation

- Code formatting and indentation is neat and consistent

You may use any of the accepted code formatting styles - eg. One True Brace Style (1TBS), the standard eclipse Java style or the supplied eclipse formatting style (which is an adaption of 1TBS), as long your code formatting is neat, consistent, and indentation / positioning of each segment of code is correct, relative to the code segments around it.

- Lines of code should generally not exceed 80 characters in length where possible, maximum line length is 100 characters.

You should endeavour to split lines which will exceed the 80-character limit by a significant amount into two or more segments where required (or think about whether you have one of the problems mentioned below), but any lines of code which will exceed the 100-character limit must be split.

Note: It is good programming practice to try to keep lines of code short, as overly long lines can be indications of A) overly verbose identifiers, B) overly complex data structures / algorithms (including nesting) or C) trying to do too much in the one statement.

- Expressions and overall source code are spaced out appropriately

Use spaces between operators and operands to make expressions more readable and use blank lines strategically to separate your code into logically related “segments”, to enhance code readability.

Use of appropriate identifiers

- Select appropriately descriptive identifiers for variables and methods which reflect the meaning of the value stored or task the method is performing to improve code readability.
- Identifiers should generally not consist of more than 3-4 “words” and, where a variable stores a value which has a specific unit of measurement, you should note that unit of measurement as the end of the identifier name (eg. tripDistanceKm).
- Avoid abbreviations (especially heavy/repeated abbreviations) where possible.
- Class names should be in title case, method and variable names

Code documentation

Comments should be included with your code as per the below:

- A “header” comment describing the purpose of the class / program
- Comments for each logically related non-trivial segment of code which describe the purpose the code segment or goal the code is trying to achieve, especially for segments which implement logical steps in the process you are implementing.
- Comments should be positioned on the line above statement(s) or structure they refer to, not on the same line.

An example of a good comment - concise and describes goal code is trying to achieve:

```
// calculate interest payable and new balance values
interest = balance + interestRate;
balance  = balance + interest;
```

Examples of poor commenting (reciting code):

```
// calculate the interest payable as the existing balance multiplied by the
```

```
// interest rate and store the result in the variable 'interest', then  
// calculate the new balance as the existing balance plus the interest  
// payable and overwrite the existing balance in the variable 'balance' with  
// the new balance
```

```
interest =      ;  
balance  =      ;
```

```
// declare an int variable called 'x' and set it to the value '1'  
int x = 1;
```