Why Good Coding Style is Important:

Coding style is important for several reasons. First of all, programs written with code are likely to be used by others beyond just the original programmer. If a program doesn’t align with good coding principles, it will take additional time and effort for those that may need to manipulate or debug the code to work out what they need to. It could even be argued that good code is important for the original programmer as after time away from working on a program, they will likely not remember exactly what it is they were doing if proper techniques aren’t used. As with everything else in programming, practice will help you understand what techniques are used to help improve coding style.

What is Good Coding Style?

While overall there are not hard/fast rules for every situation when it comes to writing quality code, there are some great guidelines to follow. Below we’ll take a look at some of the biggest issues to keep in mind when writing code.

Use of Comments:
Comments are a great way to explain code when necessary. The key to using comments focuses on the last part of the previous sentence. Only insert comments to clarify code when necessary. Your ultimate goal is to have your code stand on its own and be understandable without much explanation. Comments can be used at the beginning of a program to describe the purpose of the overall program. Comments should also be used to explain lines of code if they are confusing or irregular for some reason.

Formatting and Layout:
Programming languages have unique ways that you are required to initialize functions and what types of characters need to be used to create a program. However, there are some general formatting guidelines to keep in mind. The most important characteristic is consistency. Be consistent throughout a program in how you space within a line.

hoursWorked = (hoursPerDay * daysWorked)

Vs

hoursWorked=(hoursPerDay*daysWorked)

Both of these lines of code should work; however, the use of space in the 1st example makes the line of code easier to read.
It is also important to keep in mind the use of whitespace between lines of code. Simple use of an extra space can make a big difference in the readability of the code. It acts as a buffer between sections to help group like items together.

Another consideration in formatting is consistency of indentation. The amount of indentation should be done the same throughout the program. This again improves the overall look of the code and improves readability.

**Naming:**
Use of names for variables and functions is important. Names in your code should have meaning, but should also not be too long. For example if you were adding up a total cost of some products you might see:

```
tc = 25
```

This isn’t great because it is not descriptive. A better option would be:

```
totalcost = 25
```

This is an improvement but for readability coders often use either underscores to separate words or a concept called camelback. Camelback capitalizes every word after the first word. In this example it would be written like:

```
totalCost = 25
```

This example improves the readability of the code by capitalizing the word(s) after the first word.