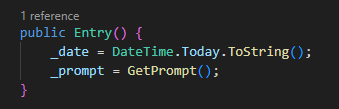
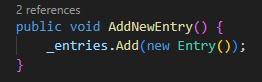
**Abstraction – Articulate**

Abstraction in programming means taking a complex concept or task, and making it simpler. If done correctly, it removes most of the time-consuming steps, and breaks it down into little pieces of code, that can be written once, and called by command later. Abstraction helps to make the code simpler to understand, and easier to maintain, because it makes the code a lot more readable. A good example of abstraction would be if you have a method to display a UI element to the screen. When you’re using an UI framework’s function to do that, you only use one command, but in fact, thousands of lines of code are running to display the element correctly. You don’t have to write those every time, you can abstract it to only one command. Here’s how I used abstraction for my assignment:

I wrote the **Journal** class**,** which has 76 lines, and the **Entry** class, which has 27 lines. One Entry Object holds its set of variables and data, and one Journal object holds many Entry objects. To Abstract this process, every time an instance of Entry is created, it pre-defines many values with its constructor:



And to create many instances of the Entry object, there is a specific function in the Journal class, that creates those instances, and automatically stores them in a list inside the object:



It gets a lot simpler to write in the main class:

