**Abstraction – Articulate**

1. Explain the meaning of Abstraction

Abstraction is making code into only what it needs to be. It takes it down to the bare bones so that it’s simple and not overly complicated.

1. Highlight a benefit of Abstraction

A benefit of abstraction is that it is easier to read because of its simplicity. If you take a break from your code or a new person looks at it will be easier to follow.

1. Provide an application of Abstraction

An example of application is creating a diagram before you begin to separate methods into different classes the best you can.

1. Use a code example of Abstraction from the program you wrote

In our Journal program we made separate classes for entry, prompt, and journal so that they could all be implemented into a single program instead of muddling it all up.

1. Thoroughly explain these concepts (this likely cannot be done in less than 100 words)

The first step when starting abstraction is preparing by categorizing what the program will need into objects/nouns. It conceptualizes what you will need to do, and these will become your new classes. After that you’ll need to define and find out what responsibility, behaviors, and what the behaviors will be stated as. In the assignment from last week, it said that thinking about a person is an abstraction because you can think about only the person and not need to actively think about the things that make them up. I like this example; it really helped me. I can bring a person to mind and not actively think about their last name, or how I met them, or every small details about their personality. This is an abstraction because I can make this separation between details and the overall idea.