Code Style

Identifiers

- Variables
    - Our variables use camel case, using names that are descriptive of what they represent without being verbose. Variables that are not simple enough to be fully described by name should have a brief single-line comment to describe their purpose.
- Class Names
    - Class names must be simple while also describing what they represent and how they are used. Class names should be distinct from each other, helping each class serve its own role. Inside the .h and .cpp files, every word within the class name should begin with a capital letter.
        - Example: “class Flower : public Plant” represents the declaration of a class called Flower that inherits from Plant. This class inherits a method to display the code to create a Flower object.
- Methods
    - Methods follow the same camel case requirements of variable names. Method names should clearly describe their functionality while remaining succinct. When possible, repeated blocks of code should be separated into private helper methods, with names that represent what functionality they are helping with.

Indenting

All code follows standard C++ indenting practices as if quickly formatted with the ctrl+i command. If method calls, statements or any other lines of code are too long to fit in one line, they are indented. The indented lines should line up uniformly. Indentation is done with tabs.

White Space

White space is added between method and function definitions. If-statement and loop logic blocks are followed with white space unless it is the end of a method or function. White space also separates operators in formulas (x * y, not x*y).

Comments

Each header file has a comment block for all its methods. Large blocks of code are explained with in-line comments. Complex variables are also explained with short comments.