Part 1: OO Concepts and Definitions – 10 points

- **Abstraction**
  - Abstraction is the concept of hiding the workings/complexity/detail of something from its user. The user is still able to use the "thing" but they do not need to know how it works, only what it might produce for them.[1]
  - We can use a class's setters as an abstraction for setting the values of various attributes in an object.
- **Encapsulation**
  - The practice of grouping together attributes and methods that operate on those attributes into a unit for the ability to decide when,how and to whom the data can be shown and accessed by.[2]
  - The getters and setters of a class are an example of encapsulation, they are methods that are bundled in a class and only work on the classes data, they also control who can access and modify the data.
- **Polymorphism**
  - The ability for something to have multiple forms. The ability for objects of different types to be accessed and modified by the same interface
  - Using compile time polymorphism in c++ with overloading functions and operators, ie, multiple instances of the same function in a class but with different input parameters
- **Coupling**
  - The degree of awareness that something has about something else, typical referred to by being tight or loose[3]
  - If two classes do not share any attributes or methods then they could be considered as loosely coupled, while a derived class is usually tightly coupled with its parent and sibling classes.
- **Cohesion**
  - The scope of operations of a class. How focused a class is on its objective.[4]
  - A class that is meant to only output a GUI but also has methods that perform numerical statistical analysis is considered to have low cohesion
- **Identity**

[1] https://stackify.com/oop-concept-abstraction/
[2] https://www.sumologic.com/glossary/encapsulation/
[3] https://www.geeksforgeeks.org/coupling-in-java/
[4] https://ducmanhphan.github.io/2019-03-23-Coupling-and-Cohension-in-OOP/

- The notion that an object is a unique “thing" even[5] if its state is the exact same as another object.
- We can have two identical objects created from the same class, but when we look closer each object has a unique space and identifier in memory. This shows that each object has a unique identity.

[5] https://atomicobject.com/resources/oo-programming/abstraction-and-identity