OOP Concept Review:

Object: An object is a software bundle of related state and behavior. Software objects are often used to model the real-world objects that you find in everyday life.

Class: A class is a blueprint or prototype from which objects are created. This section defines a class that models the state and behavior of a real-world object.

Inheritance: Classes inherit state and behavior (methods from their superclasses)

Interface: Contract between a class and the outside workd. Class implements an interface and promises to provide the behavior published by that interface.

Abstract classes: Basis for different subclasses that share behavior which does not need to be created. Subclasses must complete the behavior and have the option to override the predefined behavior as long as it is not defined final or private. Abstract methods are declared but does not contain any implementation. By using abstract classes, you can inherit the implementation of non-abstract classes. Interfaces all need implementation.

Abstraction: Process of exposing only the relevant and essential data to the users without showing unnecessary information

Polymorphism: Use an entity in multiple forms //

Encapsulation: Prevents the data from unwanted access by binding of code and the data in a single unit called object //

Inheritance: Promotes reusability of code and eliminates the use of redundant code. It is the property through which a child class obtains all the features defined in its parent class.

State: The set of values of the attributes of a particular object is called a state.

Constructor: method to initialize the state of an object and it gets involved at the time an instance of that object is created

Overloading: Different kinds and number of parameters of the same function name