Moda, Emeka A.

BSCS32E1

CS ELECTIVE 2

Prelim Assignment #1

Object-Oriented Programming (OOP) is a programming paradigm that uses “objects” to design applications and software. Here are the key principles of OOP:

Encapsulation: It’s the bundling of data, and the methods that operate on this data, into a single unit called a class. For example, a Car class might contain properties like color and speed, and methods like accelerate() and brake(). Encapsulation can make code more flexible, easy to debug, and secure by hiding the internal state of an object.

Inheritance: It allows classes to inherit features of other classes. For instance, a Truck class could inherit properties and methods from a general Vehicle class. This leads to a more organized and manageable code and eliminates the need for code duplication.

Polymorphism: It allows one interface to be used for a general class of actions. For example, the Vehicle class might have a method startEngine(). The implementation of the startEngine() method would be different for the Car class and the Motorcycle class, but an object of either class could be used to call startEngine(). This makes it easier to expand and improve software over time.

Abstraction: It’s about hiding complex details and showing only the essentials. For example, when you use a Car object, you don’t need to understand how its accelerate() method works internally; you just need to know what it does. Abstraction can help to reduce complexity and isolate impacts of changes in code.