Jason Vescas
Object-oriented Programming: CPSC 24500-001

Dr. Fadi Wedyan

4/26/2024

Assignment 8

Class and object: A class is a base to create and hold data for objects and methods, and its essentially the structure of how the code will behave. An object is an element or an instance of a class, meaning that the class will dictate what it is. the way I like to think about it is like if you were building a house. The class is the blueprints that details every thing the house should have/need, and actually building the house is the object.

class members: Class members like variables, methods, and constructors are part of a class. following the analogy I made above, the class members would be the characteristics or dimensions of the house.

Encapsulation / Information Hiding: Encapsulation is a concept that makes the data or methods within a class private. this is done to keep any external code from changing their attributes. Information Hiding is pretty much achieved through Encapsulation, as it restricts access to the data.

Generalization: In my understanding, Generalization involves making a "general" class that defines features that other "specialized" classes should use, which cuts down on reusing code.

Composition and aggregation: these are forms of association to define relationships between classes. Composition is a strong association, and aggregation is a weak association.

Dynamic allocation: refers to memory management when running code in Java. A certain amount of memory is given to the code to work with and will allocate more if needed

static Method matching: involves linking method calls to method bodies at Compile time. static methods are matched by the Compiler using the type of reference variable, and not by the type of object instance.

Dynamic Binding: allows Java to choose which method to use at runtime, which is useful when methods are overridden

Polymorphism allows Java objects within different classes to be treated like objects of a common class.

Deep copy / shallow copy: Deep copy replicates everything directly or indirectly referenced by an object, while a shallow copy replicates only an objects immediate data, but not the objects that data refers to.

fat interface: usually refers to bulky code that has too many methods that aren't all used by implementers.

Open-closed principle: states that "software should be open for extension but closed for modification". Meaning that if need be, anything that needs to be added should only be as an extension. Not directly modifying the original code

Dynamic linking and static linking: Dynamic linking loads any library code into a program when it is started, while static linking loads any library code into a program while it is compiling.

Fragile base class problem: A fragile base class is essentially where any changes made to a base class can inadvertently affect any classes that it sources data to.