Principles of Object Oriented Programming

1. Abstraction/Encapsulation
2. Inheritance
3. Polymorphism

Abstraction – Attributes and Behaviors

“Attributes and behaviors meant to represent a real world object”

“A representation of a real world item using its essential properties and actions”

Encapsulation – Separation of implementation and documentation

“Hiding of data implementation: restricting access to accessors and mutators”

Accessors – Used to ask an object about itself; its attributes/properties (e.g. get method)

Mutators – Used to modify the state of an object and/or its attributes (e.g. set method)

Advantages: The class can be changed without compromising any code that uses the class. Everything is dealt with “under the hood” so any code that uses the class doesn’t care what the class does but only cares about the result.

Inheritance – Inherits generalized parent’s use while specializing (either adding or subtracting)

“Is a” relationship

Instead of independently implementing different classes that use the same attributes or methods, you can create a base class in which the derived classes inherit all of the common properties

Question: If you have a base class that has private attributes, what’s the point of inheriting it as a protected or private parent if you lose the private attributes?

Polymorphism – Parent takes on the role of a child upon initialization

Overriding (Run-Time Polymorphism) – When a child class has a function that uses the same name as a function in its base class, but the child’s version does something different and when an object using the child class is created, the use of that function will use the child’s version.

Overloading (Compile-Time Polymorphism) – When a child class has a function that uses the same name as a function in its base class, but the child’s version has different parameters as the base class’ version and when an object using the child class is created, the function can be called with the specified parameters to use the child’s version of the function.

\_Studied 1/9

Class gets 3 defaults, default constructor, copy constructor, and destructor.

Abstract Data Types – User defined data types using multiple data types with attributes and behaviors

Redefinition, Over-riding, Overloading

Static(At compile) vs dynamic(During runtime) binding

Base Class Pointers

Function Redefinition

Over-riding/Polymorphism

Friendship

Operator Overloading

Arithmetic, Relational, Logical, I/O

Exception Handling

Function Templates

Class Templates