CA341 OO paradigm

Objects & Classes

Object-oriented programming is an extension of data abstraction. An object is a package made of private variables and public methods. There can be multiple instances of an object

Class

Member variables

Instance variables

Constructors and destructors

Constructors intialize objects.

Functions access variables and update instances.

Inheritance

Classes can have subclasses. Subclass have the member variables and method of their superclasses with additional methods.

It may override its parent class methods.

Classes can override any method that does not have a final keyword

Dynamic Dispatch

* If we declare a variable object p1 then p1 is either an object of its subclass or superclass. When a function is called for p1, the p1 object type is examined and either being of super or subclass is used.

A private component is only visible within its own class

A public component is visible anywhere in the program

A protected component is visible within its class and subclasses only.

Abstract Classes

Have no constructors

Undefined methods which are abstract

Acts as superclass to other classes which inherit

Multiple Inheritance

Multiple Inheritance is where class can have more than one superclass. The subclass inherits all member variables and methods for each superclass.

C++/Python supports MI

Java/Ada/C# do not

Mi has problems with dynamic dispatch

4 solutions to overriding methods

1. Invoke from the first listed superclass
2. Require the programmer do explicitly state which version to use
3. Same as aboe when being invoked
4. Prohibit method calls that could cause these problems.

Interface

A unit which declares operations and methods that another program must define

Interfaces define abstract methods. Interfaces avoid the problems associated with multiple inheritance.