

Exercise 1

(a) Define three classes; one class must be the base class (or parent class) and the other two classes being derived classes (or child classes). Choose from the following options for ideas or come up with your own. If you come up with your own, make sure that your assignment of the base class and derived classes pass the “is-a” relationship test as discussed in the lecture. Note that all of the following would pass this test.

- Relative (base class); Sister, Brother, Aunt, Uncle (derived classes)
- Parent (base class); Mom, Dad (derived classes)
- Appliance (base class); Stove, Refrigerator, Oven, Dishwasher (derived classes)
- Animal (base class); Dog, Cat, Hamster, Tiger (derived classes)
- Publication (base class); Book, Magazine, Newspaper (derived classes)

(b) FOR YOUR BASE CLASS, IMPLEMENT THE FOLLOWING:

- Your base class should contain at least 2 - 3 fields (attributes) with get and set methods as needed.
- Your base class must have a non-default constructor with parameters that initialize all of the fields.
- Your base class must have a **printDetails()** method that prints out a description of the object (it's fields)

(c) FOR ALL OF YOUR DERIVED CLASSES, IMPLEMENT THE FOLLOWING

- Your derived classes should contain 2 - 3 additional fields (attributes) with get and set methods as needed
- The derived class must have a non-default constructor that uses the proper syntax to call the constructor of its parent class and to initialize the new fields.
- Your derived classes should also have a **printDetails()** method that prints out the details of your ALL of the derived class attributes, including the attributes that are inherited.

(d) NOW CREATE AN ADDITIONAL CLASS (no inheritance)

- This class must have a field of type **vector** that holds pointers to your base class.
- Create a method that adds a SINGLE object to your **vector**. The object instance must be passed as a parameter to this method using a polymorphic parameter of your base class pointer type (this can hold any of the subtype instances).
- Again using the concept of a polymorphic pointer variable, write a **printAll()** method that has a for-each loop that calls the **printDetails()** methods to print the details of each object in your **vector**.

(e) IMPLEMENT A MAIN

- Write a **main** method that creates the necessary object(s), adds several instances to the list and calls the **printAll()** method. Your main must create at least 5 instances of each of your derived classes, add them to the list using your previously written method that adds pointer to base class objects to the list, and then calls the **printAll()** method.