

# COMP 1771 – Fall 2008

## Class #12: Lab Exercises

### Instructor:

Prof. Matt Rutherford – [mjr@cs.du.edu](mailto:mjr@cs.du.edu)

October 16, 2008

## Interface Mechanics

For this exercise, create an interface called `PresidentialCandidate` that defines the following methods:

**String getName()** – returns the candidate's name

**String getEndorseeName()** – returns the name of the candidate that this candidate endorses

**boolean isActive()** – returns true if the candidate is active, false otherwise

Create two implementations of the `PresidentialCandidate`:

**ActiveCandidate** – represents a candidate that is still in the race; this class has a single data member for the name that must be passed in via the constructor.

**FormerCandidate** – represents a candidate that is no longer in the race; this class has a data member for the candidate's name, and another data member for the name of the candidate that they endorse, both of which are passed via the constructor.

Create a class with a main function that does the following:

- Declares and instantiate an array of `PresidentialCandidate` objects that has the capacity for 6 objects.
- Instantiates either a `ActiveCandidate` or `FormerCandidate` using the following names (in alphabetical order):
  - Clinton, Hillary
  - Dodd, Christopher
  - Edwards, John
  - Huckabee, Mike
  - McCain, John
  - Obama, Barack
  - Paul, Ron
  - Romney, Mitt
- Calls a method `printEndorsement(PresidentialCandidate pc)` that prints something like the following:

Hello, my name is <name here> candiate for President. I am proud to endorse <endorsee here>.

or

Hello, my name is <name here> former candiate for President. I am proud to endorse <endorsee here>

## **Inheritance Mechanics**

Change the above code so that `PresidentialCandidate` is a class with the same methods. Derive `ActiveCandidate` and `FormerCandidate` from this class and eliminate any redundant methods and data members.