Assignment 2
Due Sept 15 11:59 pm

90 Points Total

Objectives

1. Use more Objective-C features - classes, protocol, extending classes, etc

Programming Problems

1. (10 points) Add the method **asDate** to the NSString class. The method **asDate** converts an NSString object to an NSDate object if the string represents a date. For example [@"8/10/2013" asDate] should return an NSDate object for August 10, 2013. (Hint: Look at NSDateFormatter class)

2. (10 points) The class method **new** on the NSDate class returns an NSDate object representing the current date and time. However this is not very obvious. Add a class method **now** to the NSDate class. The method returns an NSDate object that represents the current date and time.

3. (10 points) Add the method **yearsBetween:** to the NSDate class. The method **yearsBetween:** has one argument, an NSDate object. The method returns the number of years between the two dates.

4. (25 points) Create a Person class. A Person has a first name, last (or family) name, a birthday, and 0 or more phone numbers. The Person class should define two properties, firstName and lastName. Declare an instance variable to store the phone numbers. Normally you might declare them all the same way, but I want you to practice doing it each way. Defines static constants of type NSString for the types of phones (cell, mobile, work, home, other). The person class should have the following methods:

   + (id) firstName: (NSString *) firstName lastName: (NSString *) lastName birthday: (NSString *) birthday
           Creates a new Person object with given data

   - (NSInteger) age
           Returns the age of the person in years.

   - setPhoneNumber: (NSString *) number type: (NSString *) phoneType
           Adds a phone number of a given type.

   - (NSString*) description

Returns full name of the person, (FirstName LastName)

   - (NSString *) phoneNumber: (NSString *) phoneType
         Returns the persons phone number of the given type. Or nil if number does not
   exist.

   - (BOOL) hasNumber: (NSString *) phoneNumber
         Returns true if person has given phoneNumber.

5. (25 points) Create a ContactList class. Internally the ContactList maintains one list of your
    contacts. Your class should have at least the following methods. You may find that you
    want/need more methods in the class.

    - (void) addPerson: (Person *) newContact
          Add a Person object to the list.

    - (NSArray *) orderedByAge
          Returns an NSArray of all your contacts ordered by age, from youngest to oldest.

    - (NSArray *) orderedByName
          Returns an NSArray of all your contacts ordered by last name.

    - (NSArray *) phoneNumberFor: (NSString *) lastName
          Given the lastName return phone numbers for the first person in the list that has
    that last name. Return an empty array if no such person exists.

    - (NSString *) nameForNumber: (NSString *) phoneNumber
          Return the full name of the person with the given phone number. Return nil if no
    one has the phone number.

Grading

In addition to the points indicated for each problem there will be 10 points for style. Style in-
cludes formatting your code reasonably and consistently, using Objective C naming conven-
tions and using the appropriate language constructs.

What to Turn in

Create a Xcode project for the assignment. Use the "Command line tool" project under Mac
OS Application to create the project. In the .m file containing your main() add your two func-
tions. Include the source code to all problems in the same project. Xcode places the project in
its own directory. Place the directory (and all its contents) into a zip file. Turn in your zipped file
using blackboard under assignment 2.

If you are using the SDSU library machines to do the assignment you will not be able to use
the"Command line tool" project. Select "View-based Application" template under iOS Applica-
tion. You can then modify the main method to call your code. When you run the project the
simulator will run, but will not do anything.

Late Penalty

An assignment turned in 1-7 days late, will lose 3% of the total value of the assignment per day late. The eight day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, after the ninth day late the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.