PEER-LED TEAM LEARNING
COMPUTER SCIENCE

MEETING 12 – STUDENT VERSION
INHERITANCE AND RECURSION (CONTINUED)

BARBARA G. RYDER AND PRADIP HARI

Exercise 1. Recursion and Family Relationships
Think about your family and the relationships between its members. Imagine we have a **Person** object with fields:

mother, father : reference to **Person**
children: list of **Person**
born, died: **int** (years)
gender: "M" or "F"
name: **String**

We will use a collection of **Person** objects to represent a family. For the purposes of this simulation, we will consider all step-siblings or half-siblings to be a person's siblings (to simplify our relations).

i. Your peer leader will write someone's "family tree" on the board, trying to get back to great-grandparents and including first cousins. How can you pick a person in the family tree and determine who her mother is? Describe the operations necessary to write a **String getMothersName()** method for a **Person** object.

ii. Now select a person in the family tree and describe a **getSiblingsNames()** method that returns the names in a **String** list. Show the pseudocode for this function.

iii. Select a person in the family tree and describe a **getFemaleAncestors()** method that returns a String list of ancestor names for that specific person. Now show pseudocode for this method.

iv. Now think about a method **getMaleDescendents()** for that person that will calculate a list of male descendent names. Show pseudocode for this method. Do you see similarities with the **getFemaleAncestors()** in iiii?

v. Let's try some more interesting functions. How would you write **boolean isDescendentFrom(Person p)** which returns **true** if the receiver object is a descendent of Person p and **false** otherwise. Write the pseudocode for this function. Where is recursion used?

vi. How would you write the method **boolean isAncestorOf(Person p)** ? Show the pseudocode for this method as well.

vii. A more complicated function is **String haveCommonAncestor(Person r).** This function returns the name of **Person** q when the receiver object and r both share the same ancestor q. Write the pseudocode for this function.

Exercise 2. Inheritance

Think about your CD collection and the various kinds of CD's you own. All your CD's have a title and a recording length; you can play each of your CD's. However, there may be many genres in your collection including: movie soundtracks, live performances, books on CD, and even taped comedy performances. These different types of CD's are distinguished from one another by their attributes and by difference in behavior.

i. Think about describing your CD collection by a set of Java classes. Name the classes and describe the fields needed in each class.

ii. Do these classes have shared behaviors or shared attributes? That is, can you think of a class, related to the classes you have defined that "captures" the common aspects of CD's? Describe what fields and methods might be defined in this class.

iii. Now, can you think of ways to define subclass behavior in terms of superclass behavior in this problem. What Java construct would you use to code this relationship of behaviors?

iv. So far we have subclasses and superclasses. Can you think of a reason to define a subclass of one of your existing subclasses? Recall there needs to be something (i.e., an attribute or a behavior) that distinguishes the subclass from the superclass.

Cite this module as: Ryder, B.G., Hari, P. (2012). Peer-Led Team Learning Computer Science: Meeting 12 - Student Version; Inheritance and Recursion (Continued). Online at http://www.pltlis.org.

Originally published in *Progressions: The Peer-Led Team Learning Project Newsletter*, Volume 9, Number 1, Fall 2007.