PEER-LED TEAM LEARNING
COMPUTER SCIENCE

MEETING 8 – STUDENT VERSION
MORE ITERATION AND ARRAYS

BARBARA G. RYDER AND PRADIP HARI

Exercise 1: iPod simulation
We are going to build a class needed to simulate the workings of an MP3 player or IPod. Given:
Song class that holds the title of the song, the artist singing and the album on which the song appears, and the date of the performance;
IpodApp (an IPod application) the user interface to the song library; and
SearchType, an enumerated type including the values **song, artist, title, date,** used to specify search by-title, by-artist, or by-album

You will design and build the **Library** class, a container for a library of songs that supports adding or deleting a song and several searches of the set of songs in the Library:

- creation of a playlist of songs by a certain artist (Note: artist may be a band or group)
- creation of a playlist of songs all from one album,
- search for a particular song (irregardless of the artist)

First create the method signatures for the methods in Library, and then flowchart the **deleteSong()** and **search()** methods. Hint: think about the task of creation of a playlist of songs by a certain artist.

After flowcharting these 2 methods, write the corresponding Java code for them and test them out with the IpodApp.

Can you think of other functions you would like the IpodApp to provide? For example, think of other choices you use to create playlists you enjoy. Discuss how difficult it would be to change your classes and their methods to allow these enhancements to the software. This question is an example of how software has to evolve over time, sometimes to serve needs not anticipated by the original designers and it's good to think about this as a real issue in software development.

Exercise 2: Java Jeopardy game
Now we will play a Jeopardy game where the questions are related to Java.
There are five categories: **loops, arrays, methods, Boolean conditions, porpourri.** Each category has one wildcard question that is not about Java. We will go round-robin through the students and you can pick a category and an as yet unanswered question from that category.

Exercise 3: Search techniques – getting intuition
Linear Search. Think about how you searched for the songs and albums in Exercise 1. How could it be done more efficiently

- Suggest a list of 10 album names and assume they are placed in the Library in the order suggested.
- Pick one album and count the steps needed to find it in the list
- Now pick another album and again count the steps needed to find it in the list
- What is the maximum number of steps that could be necessary to find the specific album (e.g., 10).

Binary Search.

- Now sort the albums in your proposed Library. Try a different method of searching for an album. At each step, divide the remaining albums in ½ (approximately) and determine which ½ contains the album being searched for. Discard the uninteresting ½ and then continue this process until you find the album.
- Hand-simulate this algorithm for albums close to the end of the list and close to the beginning of the list; also do so for albums in the middle of the list. Count the number of search steps in each case and compare this to the number of albums in total.
- Can you explain why the search time (in number of comparison steps) seems to be independent of the position of the album in the Library?
- Flowchart this algorithm on the board.
- Do you see any problems that might be encountered in coding this algorithm?
- What would happen if at each stage of the algorithm we divided the remaining albums into 4 approximately equal-sized groupings? How would this affect the complexity of the algorithm?
- Now think about how you might have designed the **addSong()** method in the previous problem, so the songs in the library were sorted in alphabetical order by title. Can you see a way to do this easily? Write a flowchart of your **addSong()** method design.

Cite this module as: Ryder, B.G., Hari, P. (2012). Peer-Led Team Learning Computer Science: Meeting 8 - Student Version; More Iteration and Arrays. Online at http://www.pltlis.org. Originally published in *Progressions: The Peer-Led Team Learning Project Newsletter*, Volume 9, Number 1, Fall 2007.