Ancestor Worship in CS1: On the Primacy of Arrays

Adrienne Decker
Department of Computer Science & Engineering
University at Buffalo
adrienne@cse.buffalo.edu
The Present:
Objects-First/Objects-Early CS1

- Students should not simply be taught syntax
- Students need to problem solve
- Students need to think about design
- Students need to think object-orientedly
What History Gives Us

- **Arrays**: Contiguous, homogeneous, fixed set of data elements which can be retrieved in random-access order.

- **Why has it be traditionally taught first?**
  - Well, it was around first....
    - 1945 – Zuse’s language Plankalkül, which was never implemented
    - 1957 – First implementation in FORTRAN
    - Found its way into Algol, Pascal, Ada, BASIC, C/C++, Java
Arrays in the Curriculum

- Curriculum ’68 includes arrays in first year introduction to computing course
- Curriculum ’78 does as well
- Computing Curricula 1991 puts arrays in a knowledge unit of Basic Data Structures
- CC2001 includes arrays in the knowledge unit Fundamental Data Structures and recommends coverage for an objects-first approach to CS1-CS2
Limitations of Arrays

- Syntactic Issues – Is an array an object or not?
- Arrays are Fixed Size
- Lack of a Proper Iterator
- Architectural Issues Front and Center
- Concentrates on Mechanics Instead of Abstraction
- Leads to Misuse in Programs
Syntactic Issues – Is an array an object or not?

- Create a reference to an array.
- Create an array using `new` keyword.
- Meaningful methods on an array?

- Collections are objects and students interact with them just like all other previous objects seen during the semester.
Arrays are Fixed Size

- Can not change size over lifetime.
  - Ran out of space?
    - Create another, bigger array and move all the elements to it.
    - This leads to disaster.
  - How often do we really know the correct number of elements for our collection?
  - Collections grow as needed so there is no worry about running out of space.
Lack of a Proper Iterator

- Until 1.5, no support for iteration on arrays.
- Collections all have a method that produces an iterator, which implements the java.util.Iterator interface.
- Can iterate over elements and even remove while iterating.
Arrays are normally not implemented as classes.

Moves machine-level issues earlier to explain how the array works.

Collections allow us to focus on the higher level use of the structure, rather than the underlying implementation issues.
Concentrates on Mechanics Instead of Abstraction

- Sparsely populated arrays
- Need to distinguish between empty and used space.
- Confusing to students – focus on the mechanics of data representation rules.
- Collections allow us to focus on the programmer as a consumer of the data structure.
Leads to Misuse in Programs

- The gradebook example
- How do you keep a “row” as a “row”.
- Deleting students
- What is the mapping between a student’s name and their row in the gradebook?
- Adding students
How can we fix it?

- Notice that we normally deal with two types of collections
  - Bags
  - Associative Memory
- Students are actually quite familiar with both
  - Their backpacks/Stuff in their room
  - Dictionary/Phone Book
Problem: Game Boards

- Usually end CS1 with a game (Tetris)
- Need a board that is a 2D grid of objects
- Notice, a 2D board is simply a mapping of a pair of numbers to an element located at that position on the board – a map! (HashMap)
Position Class: The *key* to the solution

- Implements `BoardConstants`, which contains the data about how large the “pieces” of the board are.
- `hashCode()` and `equals(Object obj)` needed for using `HashMap`
Does it work?

- Two groups of students, one group that could use arrays, one group that was expressly forbidden from using arrays.

- Hypotheses (null hypothesis form):
  - There will be no difference in grades on the assignment for the two groups.
  - There will be no difference in the number of students that submitted the assignment between the two groups.
Well, it does no harm....

- No difference in assignment scores ($t = .785$, $p = .433$)
- No difference in the number of students submitting ($X^2 (1) = 1.865$, $p = .172$)
To stop the backlash....

- Arrays still belong in the curriculum
  - CS2 – show how to implement an ArrayList. Discussion of run-time analysis.
  - Many methods inside Java return arrays
  - Come as a pre-defined part of many languages
However....

- Arrays should not be the first data structure.
- Focus on design, and problem solving is so important, let’s not confuse them with mundane implementation issues so early in their careers.
- Consumer of data structures first (What are they used for? How can I use them to help me solve my problem?)
- Implementers second.