CS1

Ryan Stansifer

What is it?

- CS1 is the traditional first course for students majoring in computer science in college

- We have just heard about recent curriculum experiments with completely new computing courses.
- The CS1 curriculum has been around much longer and has evolved, too, but as it is now, it is primarily teaching Java programming.

The Curriculum

Many organizations have addressed the CS1 curriculum

1. ACM --- the Association for Computing Machinery
2. CSTA -- Computer Science Teachers Association
3. College Board (AP)
4. The International Baccalaureate (IB); Standard Level (SL), Higher Level (HL)

Schools/Individuals, too

- Florida Tech. BTW we divide CS1 into two courses CSE1001 and CSE1002
- Edgewood Jr./Sr.
- Florida Virtual School

Association for Computing Machinery (ACM)

- Members meet at the annual Special Interest Group in Computer Science Education (SIGCSE) conference or the annual Innovation and Technology in Computer Science Education (ITiCSE)
- High school teachers attend these conferences where they discuss the CS0 and CS1 curriculum issues

ACM 2008 Interim Curriculum

- The committee identified all the areas of knowledge that should be covered in four years of college and will suggest an organization into courses.
- But the 2013 report does not yet have course outlines

- There are “various possible models including imperative first, objects first, functional first, breadth first, algorithms first and hardware first. These tended to reflect different philosophies and priorities about computing, especially approaches to programming.”
- The point is colleges are teaching CS1 in completely different ways.

- The committee does articulate some of the indisputable key points.

ACM: Programming Fundamentals

- Basic syntax and semantics of higher-level language
- Variables, types, expressions, and assignment
- Simple I/O
- Conditional and iterative control structures
- Functions and parameter passing
- Structured decomposition

ACM: Problem Solving

- Problem-solving strategies
- The role of algorithms in the problem-solving process
- Implementation strategies for algorithms
- Debugging strategies
- The concept and properties of algorithms

ACM conclusion

- So, while thought-provoking, the ACM curriculum does not yet offer any specific guidance in teaching CS1 at present. We await the final 2013 report.

CSTA
Computer Science Teachers Association

- Does offer some details in their K-12 standards: http://csta.acm.org/Curriculum/sub/K12Standards.html
- This does reach the high school senior/college 1st year level, but focuses on the lower level

CS Rocks!

http://blog.acm.org/csta/

1. Write your school district administer
2. Blog
3. Network with CSTA
4. Contact your local college or university
5. Write your mayor or governor

Lisa Clayborn

CSTA meetings

- Georgia Tech. Held CSTA workshops this summer

USF. AP Summer Institute to assist AP CS teachers. *This week. Sorry!*

Florida Virtual School

1. Computer Programming 1. Python and Java
   A. http://wiki.roboteducation.org/Myro_Installation_Manual (Georgia Tech)
   B. http://coweb.cc.gatech.edu/mediaComp-teach/
2. AP Computer Science

College Board: AP Computer Science

- By far the most detailed account of CS1 comes from College Board which tries to capture what colleges and universities are doing in their first course in computer science
- Let’s take a look …

AP CS: Topics

I. Object-Oriented Program Design
   A. Program design. Understand problem description, apply data abstraction and encapsulation, class specifications and relationships, understand and implement class hierarchy
   B. Class design. Design and implement a class, choose appropriate data representation and algorithms, apply functional decomposition, use inheritance

Translation

- Java is an example of a computer programming language in which the user can compute with data stamped out from templates defined by the user. These languages are called object-oriented languages.
- Java programmers must be able to conceive and organize the data of a program into objects and write templates for their creation.
- There are many choices, and good design hides details and avoid interactions.
- A hallmark of oo languages is the ability to create similar templates from others in a process called inheritance.

AP CS: Topics

II. Program Implementation

   A. Implementation techniques. Object-oriented development, top-down development, encapsulation and information hiding, procedural abstraction

   B. Programming constructs. Primitive types vs. objects, declarations, console output, control, recursion

   C. Java library classes (limited set)

Translation

- Java knows how to do thousands of things already. For the AP test to be practical, the test makers must assume the test taker is familiar with at least some of these things -- like strings and lists. Conversely, the test taker cannot be expected to have memorized everything Java can do.
- In the fundamental core of a language are the basic constructs for controlling the execution of the program, and introducing names for values.
- Also of universal significance is the ability to create well-defined subcomputations. An important technique that causes some students confusion is subcomputations that involve themselves (recursion).

AP CS: Topics

III. Program Analysis

A. Testing

B. Debugging

C. Understand and modify existing code

D. Extend existing code using inheritance

E. Understand error handling

F. Reason about programs. Pre- and post-conditions, assertions.

G. Analysis of algorithms

H. Numerical representations and their limits

Translation

- The process of creating a program ought not to be unguided. Programs and their associated algorithms are formalizable constructs with definite properties.
- Programmers must conceive of test cases for programs, conceive of the steps of executions, conceive of the correct output according to the specification. Pre- and post-conditions articulate the properties that are necessarily true about the program.
- Programmers must have the ability to do this in the design of their own programs and in the analysis of code written by other people.
- It is important the programs take unlikely or rare situations into account.

Translation

Furthermore,

- It is important to understand how computers represent numbers as they do not possess the mathematical properties we normally take for granted.

AP CS: Topics

IV. Standard Data Structures

A. Primitive data types (int, boolean, double)

B. Classes

C. Lists

D. Arrays

Translation

- In Java one can compute with numbers (both whole and fractional) and two kinds of groups: lists and arrays.
- The boolean data type represents data with two choice: true/false or yes/no. It is named after the British mathematician George Boole.
- Java makes a clear distinction between the “primitive” types and objects.

- Notice that the AP test does not require knowledge of the “fun” media objects that are built-in to Java: sound and images.
- In fact, it requires no knowledge of input at all! (Because it is easy enough on the test to ignore input and in actual Java courses there are many ways to do input.) [I have my approach to this problem in my class, as does everybody else.]

AP CS: Topics

V. Standard Algorithms
   A. Traversals, insertions, deletions on lists and arrays
   B. Searching. Sequential, binary
   C. Sorting. Selection, insertion, mergesort

AP CS: Topics

VI. Computing in context

A. System reliability

B. Privacy

C. Legal issues and intellectual property

D. Social and ethical ramifications of computer use

AP CS: The test

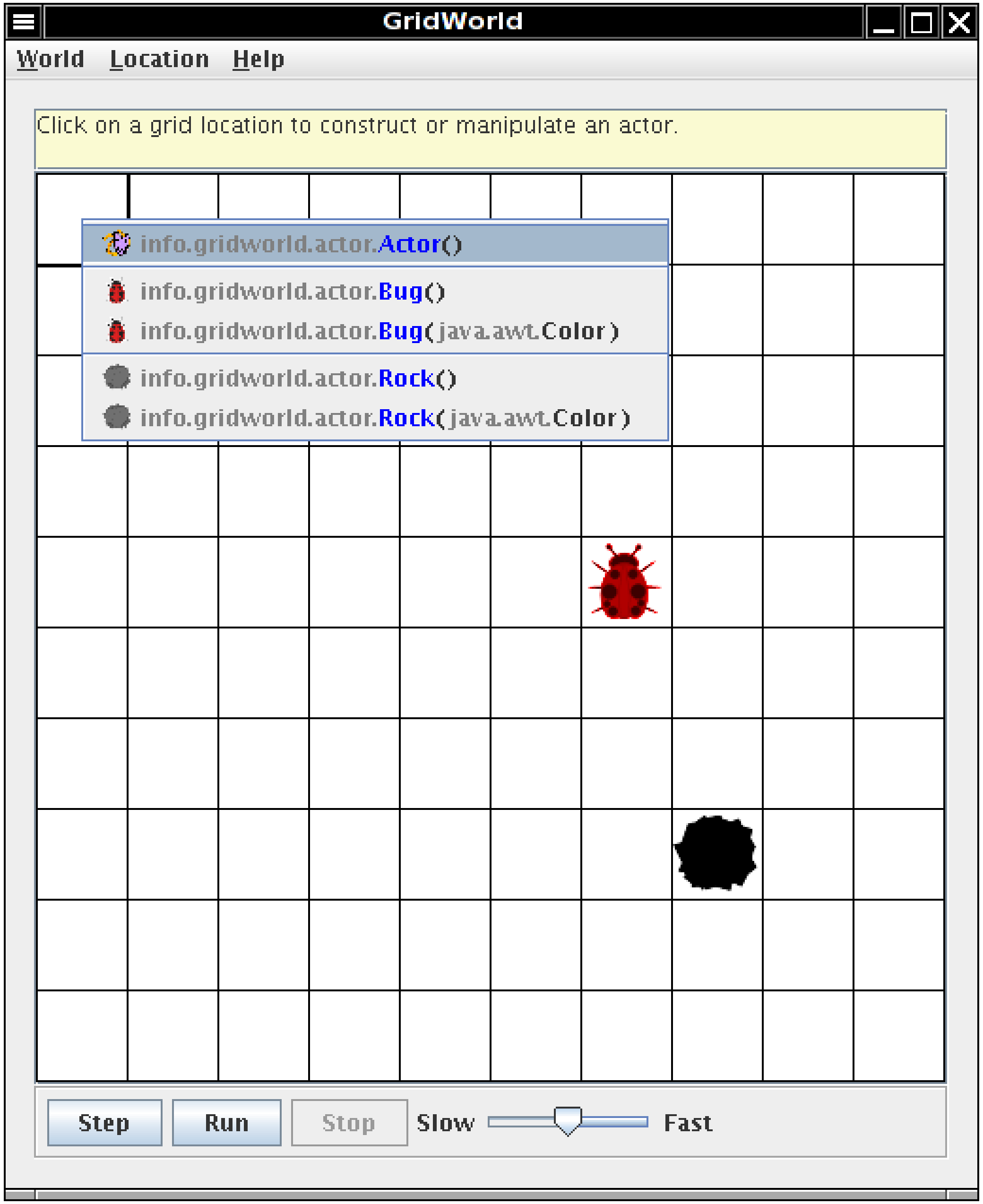
- Section I: 40 multiple-choice questions in 1 hour and 15 minutes
- Section II: 4 free-response questions in 1 hour and 45 minutes
- At least one free-response and five multiple-choice questions will be based on the GridWorld Case Study
- Given: a quick reference to standard Java interfaces; and a copy of the case study code and a quick reference to it

AP CS: GridWorld case study

Case studies are in many disciplines. They have been a part of the AP Computer Science curriculum since the 1994-95 academic year.

- Read source code written by someone else.
- Work with a program of significant length.
- Become familiar with good coding, design, and documentation practice.
- Learn about testing in a non-trivial context.
- Think through design and implementation tradeoffs.

The GridWorld case study provides a graphical environment in which students can experiment with different types of objects and observe how programming changes will affect the behavior of those objects.


GridWorld
World
Location
Help

Click on a grid location to construct or manipulate an actor.

info.gridworld.actor.Actor()
info.gridworld.actor.Bug()
info.gridworld.actor.Bug(java.awt.Color)
info.gridworld.actor.Rock()
info.gridworld.actor.Rock(java.awt.Color)
Step
Run
Stop
Slow
Fast

1. Consider the following two classes.

```
public class Dog {
   public void act () {
      System.out.print ("run");
      eat();
   }
   public void eat () { System.out.print ("eat"); }
}

public class UnderDog extends Dog {
   public void act () {
      super.act();
      System.out.print ("sleep");
   }
   public void eat () {
      super.eat();
      System.out.print ("bark");
   }
}
```

Assume that the following declaration appears in a client program.

```
   Dog fido = new UnderDog();
```

What is printed as a result of the call `fido.act()` ?

(a) `run eat`

(b) `run eat sleep`

(c) `run eat sleep bark`

[✓ (d)] `run eat bark sleep`

(e) Nothing is printed due to infinite recursion.

2. Consider the following code segment.

```
public static int mystery (int n) {
   if (n==0) return 1;
   else return 3 * mystery (n-1);
}
```

(a) 0

(b) 3

(c) 81

[✓ (d)] 243

(e) 6561

3. Consider the following code segment.

```
int[][] mat = new int[3][4];
for (int row=0; row<mat.length; row+1) {
   for (int col=0; col<mat[0].length; col++) {
      if (row<col) mat[row][col] = 1;
      else if (row==col) mat[row][col] = 2;
      else mat[row][col] = 3;
   }
}
```

What are the contents of `mat` after the code segment has been execute?

(a)
```
{{2  1  1}
 {3  2  1}
 {3  3  2}
 {3  3  3}}
```

(b)
```
{{2  3  3}
 {1  2  3}
 {1  1  2}
 {1  1  1}}
```

(c)
```
{{2  3  3  3}
 {1  2  3  3}
 {1  1  2  3}}
```

[✓ (d)]
```
{{2  1  1  1}
 {3  2  1  1}
 {3  3  2  1}}
```

(e)
```
{{1  1  1  1}
 {2  2  2  2}
 {3  3  3  3}}
```

Conclusion

- My advice is to build a CS1 course around the right set of programming projects. Fortunately, there are a lot of programming projects that have already been developed.