

Announcements

Announcements

Extra Credit Option:

Thursday Extra: Thursdays@4:00 in Science 3821

This week: Dr. Stone: “Adversarial Examples;
or, When Is a School Bus an Ostrich?”

Lab partners to be assigned over weekend for next week

email me by Sunday noon if you prefer NOT to work with
someone in the class this semester

(I will NOT honor requests to work WITH specific partners.)

Notes about Java

- To open a project in eclipse, use "New Project", not "New Java Project"
- Java/object-oriented problem solving designed to handle complexity
 - Excellent/natural for many large, complex problems
 - organize work into logical packages (e.g., stack, queue, list)
 - within package, may be multiple components (e.g., classes)
 - Safe language (e.g., bounds checking, auto. memory mgmt.)
 - Structure/safety checking
 - overhead at run time
 - code runs several times slower than C
 - as machines get larger, faster, speed may or may not be noticed
 - NOT the perspective of choice for big data, scientific computing (often too slow—use C)
 - NOT choice for much AI (want list processing, higher-order proc. in Scheme/LISP)
- Efficiency: See <https://benchmarksgame-team.pages.debian.net/benchmarksgame/faster/java.html>
 - Scheme is an interpreted language; C is compiled
 - Java has elements of both
- C vs Java: <http://www.drdobbs.com/cpp/why-code-in-c-anymore/240149452>

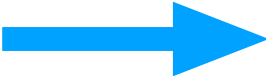
Plan for Next Week or so

- Become comfortable with basic Java syntax
- Gain understanding of simple classes, objects, variables (called fields), procedures (called methods)
- Expand syntax/concepts to consider hierarchical relationships among classes (called inheritance)
- Consider simple examples that take advantage of inheritance

For 1-2 weeks, we will need to focus on basic elements.

- Cannot cover all aspects of object-oriented problem solving at once
- With experience with syntax, basic relationships, we can start to see some payoff to problem solving and complexity
- Patience needed !!!

Today's Lab

Due Monday at start of class (1:01 pm  penalty!)

Lab requires

- some explanations (put in text file and print)
- some programs
 - write in eclipse, print code
 - programs must have main with appropriate test cases
 - include results of testing