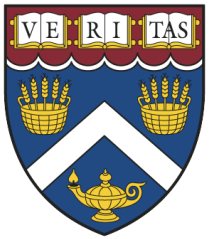
Managed Environment for the Execution of Programs

Phil McGachey

phil_mcgachey@harvard.edu



Welcome to Class

- Virtualization is a hot topic
  - Most development is done in managed languages
  - Somebody needs to make those programs run
- VMs intersect a lot of systems disciplines
- The technology's pretty cool
- Too much magic in application development…

Administrivia

- Finalized syllabus now posted
  - Check assignment dates
- Assignment Zero should be done already
- Check your prerequisites
- We'll be using a lot of Java
  - Assignments will be written in Java
  - The VM that we write will use Java semantics
  - Examples in class will assume a basic familiarity

Keeping in Touch

- In class
  - I'll be available before and after class on Mondays
- Office hours
  - Take the online poll to figure out the best times
- Email
  - I try to answer mail within 24 hours during the week
- Discussion forums
- Note that there are no formal sections
  - Take advantage of office hours and in-class discussion

Canvas

- Harvard's new Learning Management System
  - Replacing iSites for all courses over time
- Canvas will be our primary course web site
  - Assignments
  - Discussions
  - Links to videos, slides and additional material
- Make sure that you can log on
  - https://harvard.instructure.com/courses/913

Readings

- There's no assigned text book for the class
- I will be assigning readings between classes
  - Not every week
  - Generally on the order of a blog post or two
  - Related to what we discuss in class
  - Real-world examples of what we're looking at
- There will be a discussion forum for each topic

Grading

- Assignment Zero (100 points)
- Four regular assignments (200 points)
  - Work alone
- Final project (400 points)
  - Alone or in pairs
- Final Exam (500 points)
- Participation (200 points)
- Partial credit available throughout.

Assignments

- All assignments will be implemented in Java
- Assignments 1-4 are individual projects
  - Avoid discussing solution specifics on the forums
- Final project may be implemented in pairs
- Use your development environment of choice
  - Code will definitely build in Eclipse…
- Don't post code

Late Days

- Four late days during the semester
- Each late day extends the deadline by 24 hours
- Can use multiple late days on one assignment
- Late days are applied automatically

Assignment Grading

- Sample code will be provided with the VM
  - Think of it as a guide, not a verifier
  - Feel free to write your own sample code
- Grading will take two approaches
  - Running code, including the sample code provided
  - A suite of unit tests that probe fine detail
- Partial credit is available
  - I may ask you to come by or call into office hours

Final Exam

- Take-home exam
  - Released on Monday of Finals Week
  - Due by midnight on Friday of Finals Week
- Questions will be fairly in-depth
  - Answers won't be google-able
- There will be plenty of partial credit

Participation

- In-class discussions
  - Either in person or online
- Forum posts
  - Discuss the week's reading
  - Post interesting articles or examples
  - Engage in broader discussions around the topics
  - Help others with setup issues
  - Introduce yourself

Feedback

- This is a new course
  - Teething issues should be minor
  - I want to know about them when they happen
- This is also a small class
  - We can deviate if things are interesting
  - We can also slow down if things are unclear
- Some of this stuff is hard
  - Default to asking questions

Assignment Zero

- Error
  - Anything that can be raised using throw
  - Superclass of all other exceptions and errors
- Exception
  - Checked exception; must be caught or declared
  - Used for recoverable errors
- RuntimeException
  - Unchecked exception; not required to be caught
  - Generally used for bugs and fatal errors

Assignment Zero

- Static method
  - Associated with the class
  - Can't be overridden
  - No 'this' pointer
- Instance method
  - Associated with the object
  - Dynamically Dispatched
    - Runtime type of the object affects which method called
  - Has a 'this' pointer

Assignment Zero

- Processor Registers
- L1 Cache
- L2 Cache
- RAM
- SSD Disk Drive
- Magnetic Disk Drive
- Tape Storage

Assignment Zero

- malloc: Allocate a block of heap memory
  - Takes a number of bytes
  - Returns an uninitialized chunk of at least that size
- free: Return a block of memory to the heap
  - Pointer must have been returned by a malloc call
- memcpy: Copy the bytes in a piece of memory
  - Takes a source, destination and size
  - Should not be used for overlapping buffers
- memset: Fill a block of memory with a value
  - Takes a location, a value and a size

Assignment Zero

- Final methods
  - Can't be overridden
- Final fields
  - Can't be assigned to after constructor
  - Must be assigned to in all possible constructors
  - Only the field is immutable
    - Linked data (such as List contents) can still change

Assignment Zero

- Method Overloading
  - Methods are defined by class, name and signature
    - Technically also by class loader, but we'll get to that later
    - Signature contains parameters and return type
  - A method can have two attributes in common
    - Such as class and name
- Method Overriding
  - A subclass can redefine a method from its super type
  - Which method is called is determined at runtime

Assignment Zero

- Stacks are an important part of the Java VM
  - If you had trouble with the problem, let's talk
- In-place array reversal using a stack:
  - Push all values to the stack
    - If the stack is of bounded size, create an array of stacks
    - Bonus style points: use a stack of stacks
  - Pop all values from the stack
    - Insert popped values to array starting at index 0

SimpleJava

- A subset of the Java language for teaching
  - Emphasizes simplicity over features
- An accompanying virtual machine
  - Emphasizing simplicity over performance
- The target for the remaining Assignments
- More information will be on the Canvas site

The SimpleJava Language

- Inherits semantics from the Java Language Spec
- No long, short or floating point types
- Only single-dimension arrays
- No interfaces or abstract classes
- Single threaded
- No native interface
- Loads classes only from .class files; no .jars
- Very limited class library

The SimpleJava VM

- Build instructions on Canvas
- Starts out as a skeleton
  - Will be grow during the assignments
- Code refresh at the start of every assignment
  - Level field for everybody
  - Eliminates tricky bugs from previous assignments
  - Allows for bugfixes if necessary