

AP Computer Science

- Toy Night 2010
- AKA Professional Prat... Falls



Cool stuff to do with students

- Activities
- Toys
- Objets-Trouvés
- Software
- Social Interaction
- Sites Referenced
- Contact Info

Activities

*Phonebook

- * Tearing up the phone book as a method to express a binary sort. Each book can of course be used at least twice

*Paired Art (XP)

- * Students sit back to back and one is assigned to be the artist, the other to be the client. The client is given a common image and instructs the programmer how to draw it without looking at the outcome. I give about 5 minutes for the first iteration and then allow students to do it three more times so each student can have two chances at each task. Great images from pop culture include Batman, Superman, McDonalds, Pepsi, Transformers....

Manipulative Activities

*Sandwiches

- * Students write down exactly what they need to do to make the sandwich and then follow said algorithm for making the sandwiches.

*Paper airplanes

- * Algorithms and development again with no food allergies :P
- * Each student writes the instructions for making a paper airplane and has another student follow them. Grade planes based on success

More Manipulatives

*Snowflake

- * Students create a snowflake and instructions and have peers try to follow instructions and get the same snowflake

*T-Shirt

- * XXL T-Shirt and have students program one another to put it on.

Actual physical stuff works too?

*Tie shoes

- * Write down their algorithm and have another student follow it.
- * Great to record as homework and then have students show in class so they can see other algorithms in action.

*Sorting Legos

- * By color, size, shape...
- * More tactile than some other activities.

Activities

*Chocolate Bars

- * Breaking a Hershey style bar into multiple pieces to show how many pieces – You count your piece and report to calling student to build up the answer: recursive solution.

*Cell phone math

- * Recursion via calling friends on their phones to get the sum of numbers up to 6, boss had called to ask for original number

Manipulative Action

★ Binary math

- ★ Students stand in two rows with their arms outstretched. When they turn and thus “carry” over and cause their neighbors value to change it demonstrates the steps in binary addition.

★ Scavenger hunt – linked lists

- ★ Students have a clue(pointer) and a prize(value) that can easily demonstrate the logical structure. Say a detour or change needs to be made and you send the students out to change the clue/prize so traversing the list is grokked.

More and more

- * Magic Cards 0/1 – use for binary math (cs unplugged)
- * Tangrams
 - * How to make various shapes – Algorithm Design/Execution
- * Post-its and references
 - * Use the post-it to show where the objects are and what points to them
- * Candy and main method, pass the candy or just pass the name
- * Physical factorial
 - * Students come up to the board who know the step for n , calls up $n-1$ and $n-2$...

More Activities

- ★ Merge sort – doc projector or oversized cards
 - ★ Three steps to do the sort
 - ★ Karate chop with the document projector
 - ★ Instruct and ensure students close their eyes
 - ★ (make changes)
 - ★ Open eyes
 - ★ No document projector – even better. Have your students standing holding the oversized cards so others can see and use a sword as the dividing mechanism! (freak out students at the sort of 3)
- ★ Ask for five dollars
 - ★ Point to it vs hold it...
 - ★ Who has the best deal?

Toys

*Monkeys

- * Direct relation to linked lists
- * Standard jar, keychain, fisher price myFirstMonkeyJar
- * Implementation shows students the need to refer to proper parts of the list and how to xfer or add etc
 - * Keychain monkeys make a great grad/post final gift – Oriental Trading Co.

*Dixie cups and cup dispenser

- * Stacks and Queues
- * Especially good if you have the opaque cups so you can really visualize the peek method
- * Write the number or some data on the cup and then :)
 - * With a removable top you can also model this as a Queue

Manipulatives

★Pez!!!

- ★ The Tastier implementation of a stack

- ★ Push/Pop and Peak are all available

 - ★ FYI sour pez are the best

- ★ Cool toys available at <http://woodgears.ca>

★Puppets as a teaching tool

- ★ Googly eyes monster finger puppets as iterators

- ★ Concrete enhancement of abstract concept

- ★ The funky strange thing that goes on the end of your finger as a reference for the “funky” attachment to a collection

- ★ You must talk to the googly monster not to the person wearing it so students can understand the importance of manipulating the structure via the appropriate tool rather than manipulating the data itself.

Objets-Trouvés

*Found objects :P

- * Little kids' things / Kitchen Items

- * Cups

- * Teaspoons

- * Shapes!

- * Nesting Sorting cups

- * Have students show you

*Decks of cards

- * Sort

- * You don't say how to sort (ace, king, bridge style)

- * Kids freak and have to find a way to sort

More Found Things

- ★Cards – ArrayLists

- ★Dr Seuss in addition to being a writer is an early computer scientist

 - ★ Yertle the turtle

 - ★ Cat in hat comes back..

 - ★ King louie cats, 1001 hats

 - ★ barthomew

- ★Office depot – business card magnets and sticky backs for cards

- ★Poker! As a case study

- ★Oriental Noodle box – Linked Lists

Lost and Found

*Massive parallel sort of decks of cards

- * Pass out a large multi-deck stack of cards and get the kids to sort them together
- * Encourage them to get it done
- * Look mine is sorted, hurry and get yours done
 - * Yours is already sorted of course :P
- * Which is the best way to sort, lots of implementation options
- * Search too

*Grandma's pill box

- * SMTWTFS
- * ArrayLists

Did you find my missing ...

*Event driven programming

- * Throwing kooshes as a event and listener example
- * Student A is a koosh listener
 - * Waits for a koosh
- * Student B is a koosh actor
 - * Tosses the koosh

*Bugbear

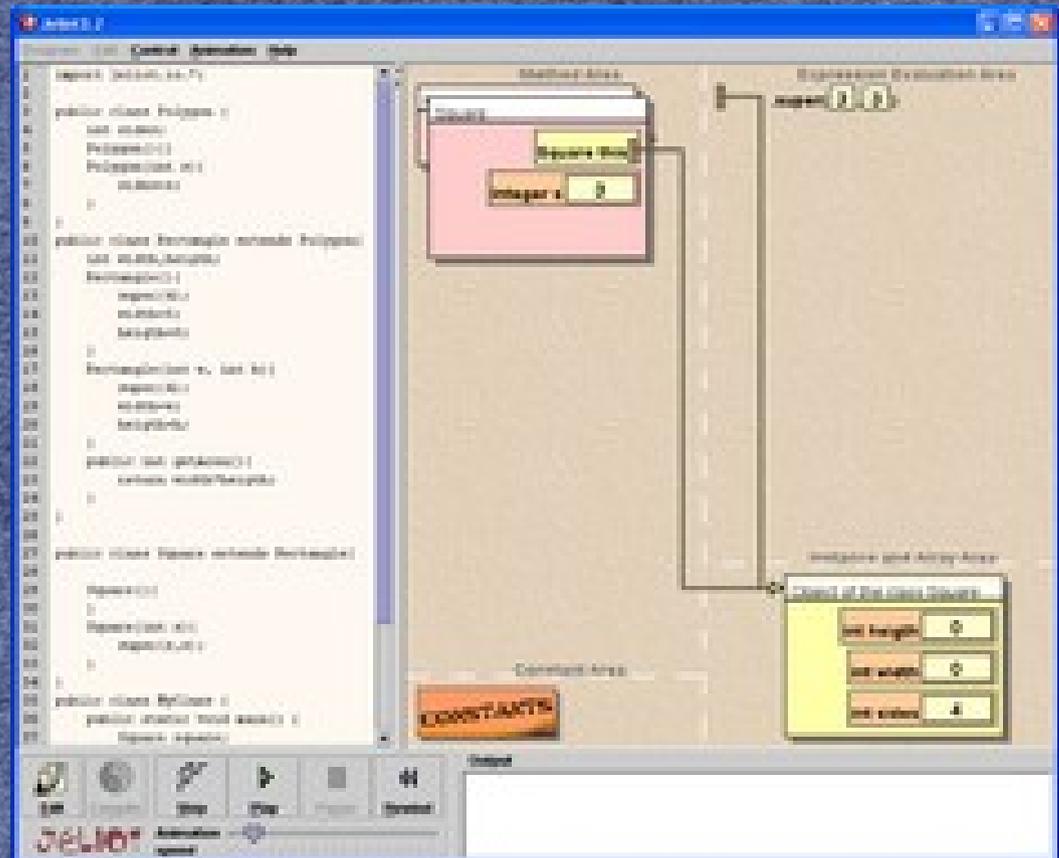
- * Teddy bear sits in front of the code – talk to the bear and you find the bug – proxy learning

*Students sort selves based on height

- * Who will be the comparator?
- * Suggest you, the teacher

Software

- ★ Jeliot – editor/visualizer recursion is nifty – graphics that show all calls and returns
 - ★ Static v instance
 - ★ Plug in for moodle



JavaWIDE

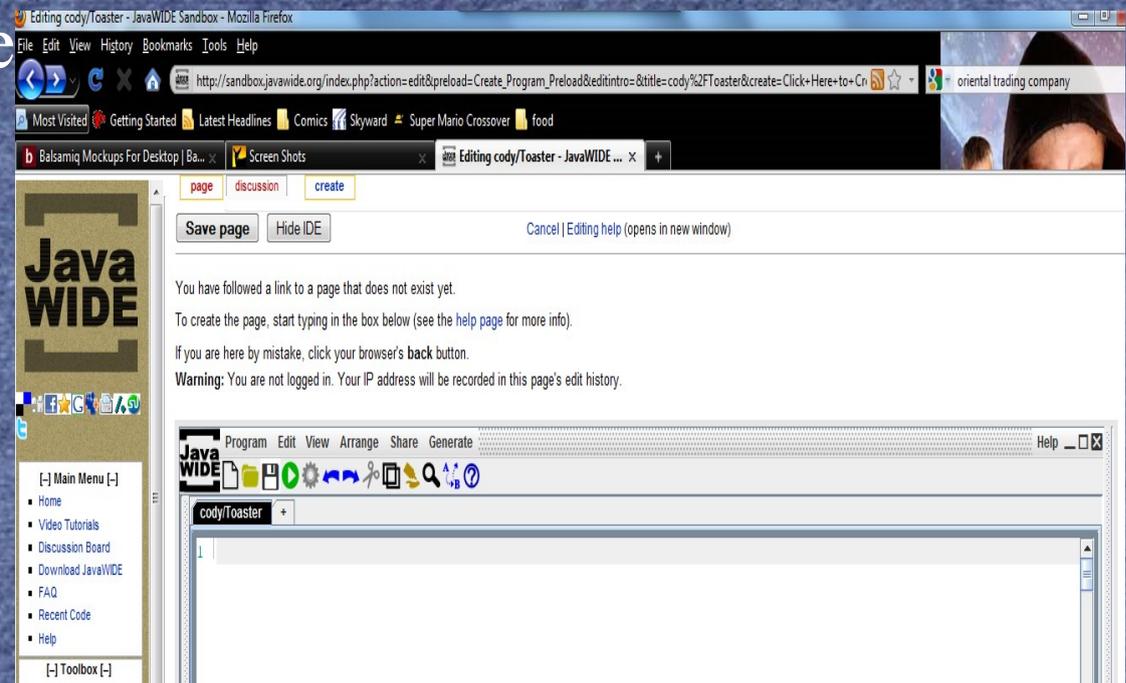
★ Wiki Integrated Development Environment

★ History viewer! You can view the changes easily in a web environment

★ Environments include:

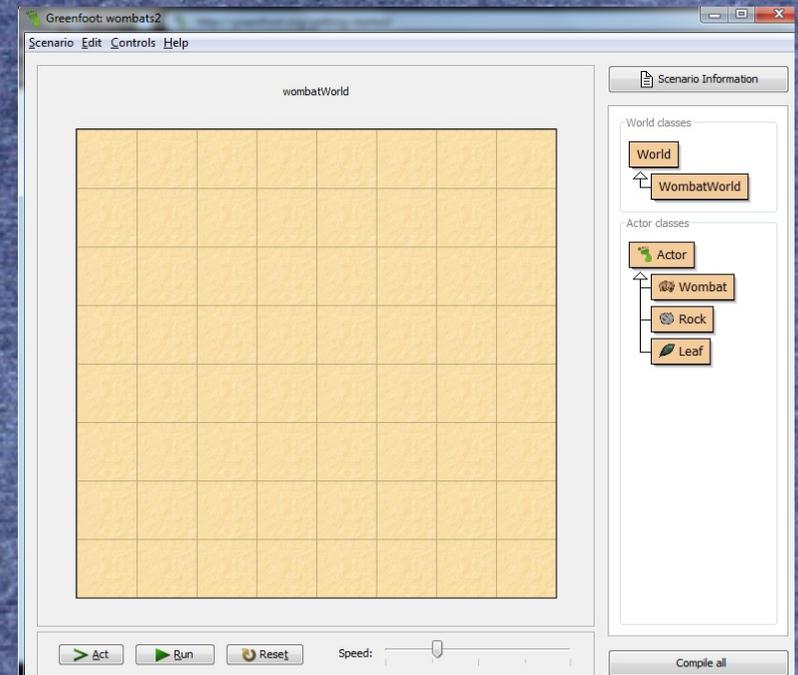
★ Game engine

★ Gridworld



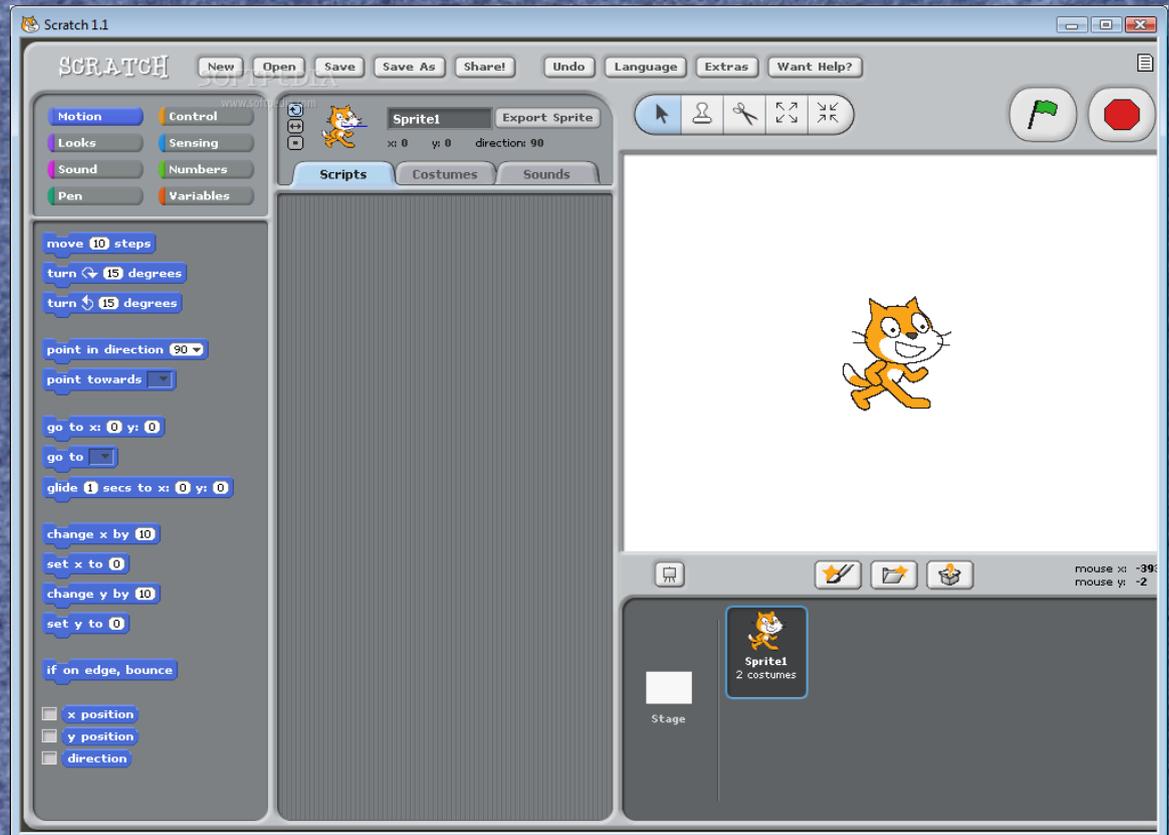
Greenfoot

- * Object Oriented development environment that is similar in structure to BlueJ
- * Designed to operate in a 2D grid based environment for age 13+ students.
- * Very visual and easy to setup and use



Scratch – and Scratch BYOB

- *Not beer but blocks
- *Similar structure to the block style language for android
- *Implementation of the lambda calculus
- *Panther



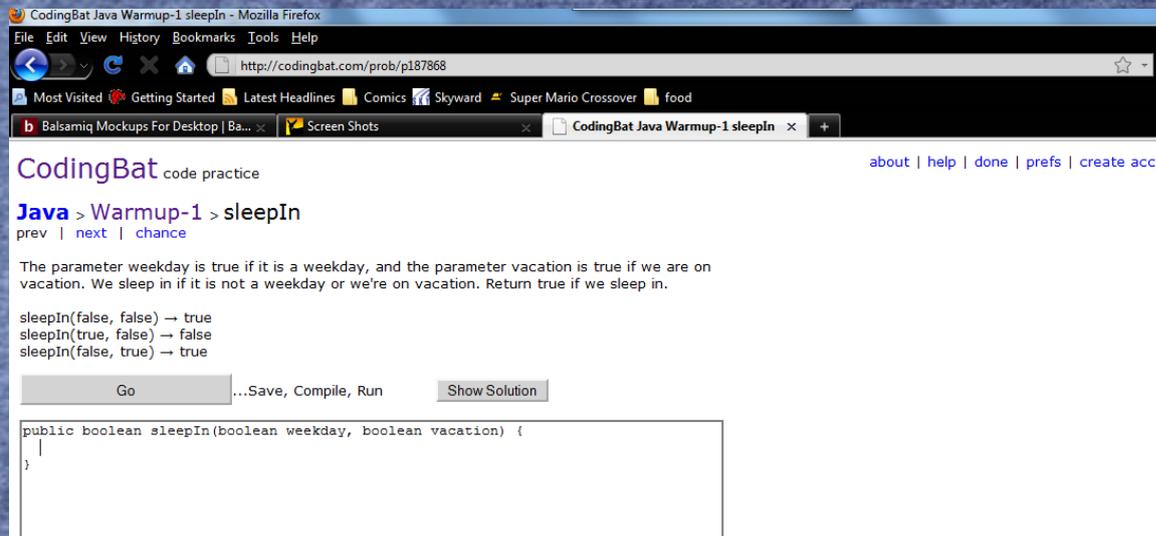
*JavaBat = CodingBat

*Python too

*Little problems that can run in realtime on web as test cases

*Display on the browser

*The site does presume a pedagogy order



The screenshot shows a web browser window with the URL `http://codingbat.com/prob/p187868`. The page title is "CodingBat Java Warmup-1 sleepIn - Mozilla Firefox". The browser's address bar and tabs are visible. The main content area displays the CodingBat logo and navigation links: [about](#) | [help](#) | [done](#) | [prefs](#) | [create account](#). Below this, the problem title is "Java > Warmup-1 > sleepIn", with sub-links for [prev](#) | [next](#) | [chance](#). The problem description states: "The parameter weekday is true if it is a weekday, and the parameter vacation is true if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return true if we sleep in." Below the description are three test cases: `sleepIn(false, false) → true`, `sleepIn(true, false) → false`, and `sleepIn(false, true) → true`. At the bottom, there are three buttons: "Go", "...Save, Compile, Run", and "Show Solution". A code editor area contains the following Java code snippet:

```
public boolean sleepIn(boolean weekday, boolean vacation) {  
    |  
    }  
}
```

*Software

*PracticeIt

- * University of Washington

*Lon – Steganography

- * Pic viewer and hider allows for easy visualization of the reality and then allows for more complex implementation

*Software²

- *Chromakey/greenscreen

- *Take students picture and paint them into a fun environment, gives them the opportunity to do even more manipulation as they like

- *graham@grahammitchell.com

- *More info available after we get home

- *Paint over most in 255 green or another color of your choice

- *The code takes ther RGB that is not green and puts them together

- *20 lines of code

- *Banner paper as a source

- *Broadcasting teacher if they have a greenscreen

*Software³

- *GUI – Balsamiq

- *Mock up a gui

- *How to make a good UI – but there is no code behind it but more the visual expansion

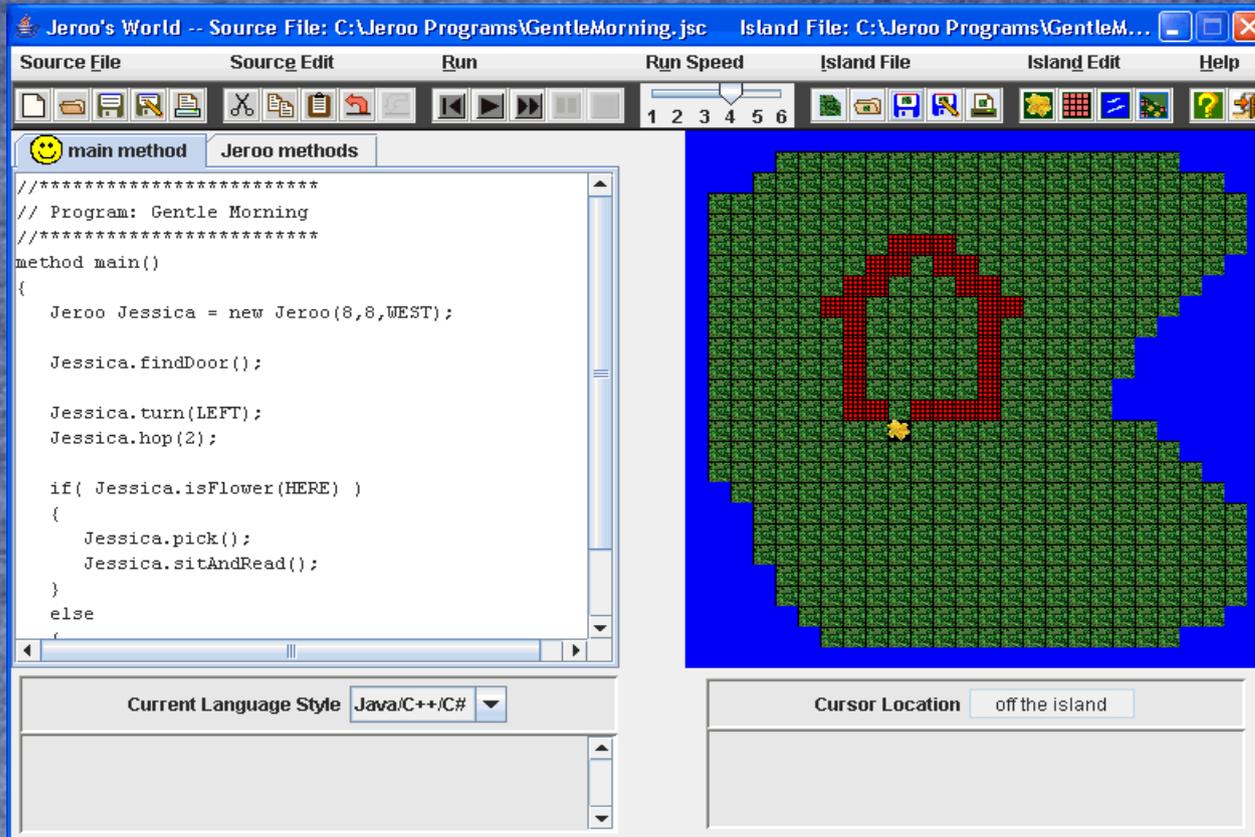
- *Free license for high schools, see contact page

- *Sample mockups also available.

*Jeroo

*Display output and mapped code

*Multiple languages including Python c++ VB java



*Gridworld extension

*A simple user interface I developed for writing simple grid-based games using the AP Computer Science's core GridWorld classes. My GridDisplay class provides support for displaying objects as images on a background of colored cells, and for handling keyboard and mouse input. My Sprite class has been adapted from GridWorld's Actor class. Unlike an Actor, a Sprite knows the name of the image file used to display it, but does not have a color or direction. A Sprite cannot be asked to act. My GridDisplay is useful for teaching the Location and Grid types in a meaningful context, long before introducing Bugs and Critters. A sample game and quick reference information are included in the .zip file.

*Social Interaction

*Don't Be Creepy

*Guidelines for social interaction for those who may not quite get how to interact as well with others as could be preferred.

*How to avoid looking like a stalker

*How to use tact effectively

Sites Referenced

- <http://dontbecreepy.com>
 - How not to be creepy
 - you can also become a facebook fan
- <http://www.cs.cmu.edu/~fberg/projects/APGridDisplay.zip>
 - Dave's gridworld extension
- <http://woodgears.ca>
 - CS Unplugged
- <http://javawide.org>
 - WIKI development environment

More Sites

- <http://scratched.mit.edu>
 - Scratch and Scratch BYOB
- <http://www.orientaltrading.com>
 - Cheap and fun stuff
- <http://webster.cs.washington.edu:8080/practiceit/>
 - Practice It! – online Java resource

Contact Info

- Thanks to all who presented and shared, the entire community will appreciate it!