

Today's "Menu"

(week 12)

- Announcements
- Intro to OOP terminology and concepts
- Exam Q & A
- Questions?

15-Apr-10

COMP 480 - Winter 2010

1

Announcements

- Exam period: Wednesday – Saturday

15-Apr-10

COMP 480 - Winter 2010

2

OOP

- Object = code + data
 - “bundle” data and operations on the data in one reusable unit.
 - Attributes (**variables**) and Actions (**methods**)
- Object Oriented Program – use and create objects.
- Easy to re-use
- You’ve used objects, e.g., Image, list etc.

15-Apr-10

COMP 480 - Winter 2010

3

Basic Terminology

- **Class** – ‘abstract’, ‘cookie-cutter’, ‘blue-print’
- **Instance** – a real ‘instance’ of the class.
 - E.g., Class Student, real Student Stacey
- **Constructor** (special method)
 - “Automagically” initialize instance
- **Methods**
 - Functions associated with Objects
 - **Setter/Getter** (Mutator/Accessor)

15-Apr-10

COMP 480 - Winter 2010

4

Example

- Person Class:
 - Think attributes
 - Behavior
- Bank Account Class:
 - Attributes (state variables)
 - Behaviors (methods)

15-Apr-10

COMP 480 - Winter 2010

5

- **Inheritance** – ‘inherit’ code and data
 - Hierarchy of objects
 - Linked via “is-a” relationships
 - E.g., **Superclass** Person, **subclass** Student (a student has all of the attributes of a Person – so *is a* specialization)
- Single vs Multiple Inheritance
 - Person -> Student
 - Truck
 - Car – SUV inherits from both

15-Apr-10

COMP 480 - Winter 2010

6

Other CS Topics

- TDD – Test Driven Design
 - How to design programs with testing in mind
- Data Structures
 - How to effectively store data
 - E.g., we used variables and lists
 - But there are: stacks, queues, trees, heaps, arrays, linked lists
- Algorithms (recursive and iterative)
 - Searching
 - Sorting
 - Analysis (“Big-O” complexity measures)

15-Apr-10

COMP 480 - Winter 2010

7

Tools

- IDEs + Editors
 - Explore your tools!
- API
 - Explore your tools!
- Debuggers
 - Inspect variables in lieu of ‘print’ statements
- Profilers
 - Measure the ‘speed’ of your code, locate bottle necks

15-Apr-10

COMP 480 - Winter 2010

8

Other Languages

- Most likely
 - Will require **compile** stage
 - Will require **type** specification
 - More **verbose** – but not more capable
 - More **complex** (e.g., IO) – but not more capable
- Java - very popular and portable
- C/C++ - popular and somewhat portable
- Perl - yuck! Ruby .. neat. Python ... :-)

15-Apr-10

COMP 480 - Winter 2010

9

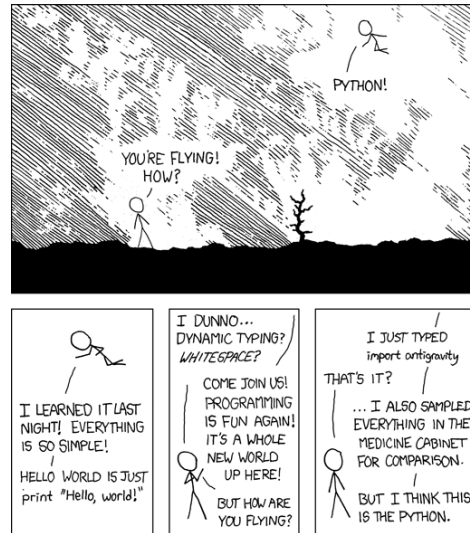
Summary/Recap

- Questions?

15-Apr-10

COMP 480 - Winter 2010

10



15-Apr-10

COMP 480 - Fall 2009

11