CSC 102
INTRO TO PROGRAMMING WITH PYTHON

REVIEW

MICHAEL GROSSBERG
MATERIAL

☐ EVERYTHING

☐ LECTURES

☐ HOMEWORKS

☐ TEXTBOOK (ALL OF IT)

☐ MOSTLY PYTHON 2.6

☐ SOME C++

☐ POTENTIALLY: TRANSLATE PYTHON PROGRAM TO C++
FORMAT

☐ LIKE MIDTERM

☐ PART I: WRITTEN, NO COMPUTERS, NO REFERENCE

☐ PART II: PROGRAMMING, WE SUPPLY REFERENCES

☐ ALL CLOSED BOOK

☐ NO NOTES

☐ NO CELL PHONES

Tuesday, December 14, 2010
PART I: WRITTEN FORMAT

- PART I: WRITTEN, NO COMPUTERS
  - (A) DEFINITIONS
  - (B) SHORT ANSWER CONCEPTS
  - (C) FIND THE BUGS 1 OR 2
PART II: PROGRAMMING FORMAT

☐ ONLY REFERENCES

☐ YOU WILL HAVE PYTHON STANDARD DOCS

☐ NUMPY/SCIPY/MATPLOTLIB (IF REQUIRED)

☐ WWW.CPPREFERENCE.COM DOCS FOR C++

☐ (A) WRITE SOME FUNCTION (MUST PASS TESTS)

☐ (B) WRITE SOME PROGRAMS (MUST PASS TESTS)
WHAT FOLLOWS IS **NOT** COMPLETE

- **Just some questions to get you started**
- **Understanding means being able to explain**
- **You cannot learn just from reading you must**
  - **Program**
  - **Write**

Tuesday, December 14, 2010
SOME CONCEPT REVIEWS

- INTERPRETED LANGUAGE
- COMPILED LANGUAGE

- WHAT KIND OF LANGUAGE (COMPILED OR INTERPRETED) IS C, C++, C#, JAVA, PYTHON?
- WHATS GOOD ABOUT C++? WHEN WOULD YOU USE IT?
- WHATS GOOD ABOUT JAVA? WHEN WOULD YOU USE IT?
- WHATS GOOD ABOUT PYTHON? WHEN WOULD YOU USE IT?
WHAT'S A VARIABLE?

WHAT'S A TYPE?

X = 5

WHAT'S THE NAME OF THE VARIABLE?

WHAT'S ITS VALUE?

WHAT'S ITS TYPE?
NUMERIC TYPES PYTHON SUPPORTS

- INT
- FLOAT
- COMPLEX
- LONG (INT)
OPERATORS

- **BINARY**
  - +, =, *, /, %

- **ASSIGNMENT**
  - =, +=, -=, /=, *=

Tuesday, December 14, 2010
BOOLEAN AND LOGIC

□ TRUE/FALSE

□ AND/OR/NOT

□ >, >=, ==, !=, <, <=, AND MORE
CONTAINERS

- Lists
- Tuples
- Sets
- Dictionaries

What are they?

Common points?

Differences?

What can they hold?

When do you use one over another?
STRING OPERATIONS

Are you a string ninja?

Can you search/replace?

Can you go back and forth between lists and strings: join and split?

Can you format types using strings?

Center/justify/padding/precision
CONTROL STRUCTURES

- CONDITIONALS: IF/ELIF/ELSE

- LOOPS:
  - FOR/ELSE
  - WHILE/ELSE
  - BREAK/CONTINUE
USER INPUT/OUTPUT

- INPUT:
  - RAW_INPUT, INPUT

- OUTPUT:
  - PRINT
  - SYSSTDOUT.WRITE
  - SYSSTDERR.WRITE
FILES

☐ OPEN FILE FOR

☐ READING

☐ WRITING

☐ READ/WRITE

☐ CLOSE FILE
FUNCTIONS

- FUNCTION INPUT
- FUNCTION ARGUMENTS
- PASS BY REFERENCE/VALUE
- FUNCTION OUTPUT
- RETURN
- SCOPE OF VARIABLES
REFERENCES

- Pass by references
- Pass by value
- Containers hold references
- Deep copy
- Shallow copy
RECURSION

☐ THERE WILL BE A RECURSION QUESTION

☐ LOOK OVER THE LECTURE/READ THE BOOK
EXCEPTIONS

- TRY
- EXCEPT
- FINALLY
- RAISE
- WRITING YOUR OWN WITH THE EXCEPTION CLASS
IMPORT TO GET LIBRARIES

☐ IMPORT FOR LIBRARIES
<table>
<thead>
<tr>
<th>Classes</th>
<th>Bank Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle</td>
<td>Data</td>
</tr>
<tr>
<td>Data</td>
<td>Actions</td>
</tr>
<tr>
<td>Actions</td>
<td>Data:</td>
</tr>
<tr>
<td></td>
<td>Balance</td>
</tr>
<tr>
<td></td>
<td>Owner</td>
</tr>
<tr>
<td></td>
<td>Actions:</td>
</tr>
<tr>
<td></td>
<td>Deposit</td>
</tr>
<tr>
<td></td>
<td>Withdraw</td>
</tr>
</tbody>
</table>

What's an object as opposed to a class?
CONSTRUCTING CLASSES

- CONSTRUCT WITH __INIT__
- NEED SELF ARGUMENT IN EVERY METHOD
- SELF.X STORES VARIABLE X
OVERLOADING FUNCTIONS

- SQUARE (NUM)
- SQUARE (STRING)
OVERLOADING OPERATORS

☐ __STR__

☐ __ADD__ AND TONS MORE!

☐ __SUB__
OBJECT ORIENTED PROGRAMMING

- Abstraction (Data + Action)
- Encapsulation (Outside sees only need to know)
- Inheritance (Extends functionality)
- vs. Composition (Has A/Is A)
- Polymorphism (Like overloading)
C++ VS PYTHON

- C++ has more syntax
- Curly braces for blocks
- Need to declare every variable
- Everything I just said about Python in C++ too

Know translation
C++ VS PYTHON EXAMPLE

- C++ object vars/methods are by default private
- Python object vars/methods by default public
- C++ you use "public:" section to make things public
- Python you use "_var_method" to make private
- Safe C++ uses getters/setters
- Safe Python uses properties with setters/getters
BEYOND THE COURSE

☐ TIPS: IF YOU WANT TO DEVELOP SOFTWARE

☐ KEEP UP WITH THE DEVELOPER COMMUNITY

☐ HTTP://WWW.DZONE.COM

☐ HTTP://RADAR.OREILLY.COM/

☐ HTTP://WWW.IBM.COM/DEVELOPERWORKS

☐ GOOGLE THESE: ALLTHINGSD, TECHMEME, SLASHDOT, READWRITEWEB, ALLY INSIDER, TECHCRUNCH, HACKER NEWS, SMASHING MAGAZINE

☐ WRITE CODE!
THANKS