CSCI 1300

Intro to Computing

Gabe Johnson

Lecture 8 Feb 1, 2013

Python Ranges

Upcoming Test

Test 1 **Friday, Feb 8**

Test 1 covers all Python concepts so far

I am releasing a ‘cheat sheet’ for this test. It contains a quick reference of all the concepts we’ve covered, and several questions that will appear on the test in some slightly alternate form.

Lecture Goals

1. Announcements
2. Python Ranges
3. More Python Lists
4. Questions about HW2?
5. Test Cheat Sheet

Announcements

The errors are:

g++ -Wall -o "binary_search_tree" "binary_search_tree.cpp" (in directory: /home/user/Documents/Homework 2)
/usr/bin/ld: /usr/lib/debug/usr/lib/i386-linux-gnu/crt1.o(.debug_info): relocation 0 has invalid symbol index 11
/usr/bin/ld: /usr/lib/debug/usr/lib/i386-linux-gnu/crt1.o(.debug_info): relocation 1 has invalid symbol index 12
/usr/bin/ld: /usr/lib/debug/usr/lib/i386-linux-gnu/crt1.o(.debug_info): relocation 2 has invalid symbol index 2
/usr/bin/ld: /usr/lib/debug/usr/lib/i386-linux-gnu/crt1.o(.debug_info): relocation 3 has invalid symbol index 2
/usr/bin/ld: /usr/lib/debug/usr/lib/i386-linux-gnu/crt1.o(.debug_info): relocation 4 has invalid symbol index 11
(and so on)
/usr/lib/gcc/i686-linux-gnu/4.6/../../../i386-linux-gnu/crt1.o: In function `_start':
(.text+0x18): undefined reference to `main'
/tmp/ccfRvWhM.o: In function `insert_data(bt_node**, int)':
binary_search_tree.cpp:(.text+0xd2): undefined reference to `insert(bt_node**, bt_node*)'
collect2: ld returned 1 exit status
Compilation failed.

This means you have stale object files. ‘make clean’ and then ‘make’ to refresh everything.

Announcements

```
$ python basic_functions.py
  File "basic_functions.py", line 31
    return 4
    ^
IndentationError: unexpected indent
```

This means you (or your editor) have inserted spaces when the interpreter expected tabs. Only defense against this is to be really careful, and to compile/interpret frequently.

Announcements

By the way, you can compile a Python script to check for syntax errors *without running the program*:

```
python -m py_compile basic_functions.py
```

We’re On Teh Twitters

There needs to be a quick way of getting info out about non-critical info about the class, or about Retrograde, or little helpful hints about homework.

Follow **@cs1300_cs2270** for updates.

I’ll still send email if something major happens, so this isn’t required.

Python Ranges are Lists

A ‘range’ is really just a list:

```
>>> x = range(3, 10)
>>> print x
[3, 4, 5, 6, 7, 8, 9]
>>> type(x)
<type 'list'>
```

Building a Range

Build a range with the aptly-named ‘range’ function:

range(6) → [0, 1, 2, 3, 4, 5]

this makes the range from 0 up to but not including 6

range(3, 10) → [3, 4, 5, 6, 7, 8, 9]

this makes the range from 3 up to but not including 10

range(3, 20, 4) → [3, 7, 11, 15, 19]

this makes the range from 3 up to but not including 20, incrementing by 4 each time

Iterating with Ranges

Iterating on a range is *exactly the same* as iterating on a List, because the range() function returns a List.

These have the same output:

```
print “Starting...”
for i in range(4):
   print i
print “Done”
```

```
print “Starting...”
print 0
print 1
print 2
print 3
print “Done”
```

Doing Stuff With Lists

my_list = [ ]
other_list = range(3, 20, 4) → [3, 7, 11, 15, 19]
my_list.append(11) → [11]
my_list.insert(20, 4) → [20, 11]
my_list.extend(other_list) → [20, 11, 3, 7, 11, 15, 19]

my_list.index(7) → 3
my_list.index(999) → ValueError :(
my_list.count(11) → 2
my_list.sort() → [3, 7, 11, 11, 15, 19, 20]
my_list.remove(15) → [3, 7, 11, 11, 19, 20]
my_list.pop() → returns 20, list is now [3, 7, 11, 11, 19]

Homework 2 Questions?

I can break out my editor and
code up some examples.

Test 1: Next Friday!

Cheat Sheet Available soon.

Watch Github, follow @cs1300_cs2270 to be notified of updates. I will update this document as time goes on to clarify student questions.