CSCI 4500/6500: Programming Languages

Course Overview

Short Term Plan

- **Today go over expectations and course plan**
- **Next week introduction to programming languages**
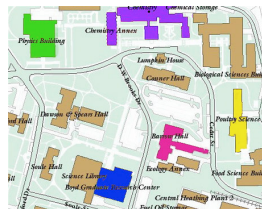- **Next week also discuss presentation topics & some advice**

Administration / Logistics

- **Who am I?**
  - » Office: Boyd 219C
- **Class:**
  - » Boyd 306
- **maria@cs.uga.edu**
- **Office Hours: Wednesdays 1:00-2:30 pm**
  - » And by e-mail appointment
- **TA: TBD - check class web page for updates…**

Communication

Web Page:
  www.cs.uga.edu/~maria/
  » **classes/4500-Spring-2012/**

- **Your Responsibility**
  - » **Understand policies, honor code**
  - » **Work independently on projects/hw**
  - » **Check page often for updates "refresh" to get latest copy**

Email list:
- **CS-PL@listserv.uga.edu**

Course Objective

- **Exposure to different programming languages**
  - » **Understand various language constructs and meaning**
    - – **ALGOL, C, Java, Lisp, Scheme, ML, python, ruby (maybe), Prolog**
- **Build appreciation for valuable language features**
- **Improve your background when choosing a language to program in**
- **Increase your ability to learn a new language**
- **Introduction to research on programming languages, past and present.**

How we're going to do it

- **Read & Listen**
  - » **Required:**
    - – **Programming Language Pragmatics, 2/e (or later), Michael L. Scott, University of Rochester (3rd version or later)**
    - – **Technical papers (summaries and presentation) (weekly)**
  - » **Optional:**
    - – **Concepts of Programming Languages, 7/e, Robert W. Sebesta, University of Colorado, Colorado Springs**
- **Practice**
  - » **5-6 programming assignments**
- **Test**
  - » **2 Midterms, 1 Final, Quizzes**
- **Talk and think in class, and outside!**
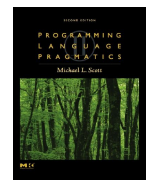
- **Theory 45%**
  - » 2 Exams (10% each) + Final 15% + Quizzes 05% = 40%
- **Practice 60%**
  - » Homework, weekly summaries & presentation & programming assignments & session chairing
- **Participation 5%**
  - » 100% attendance will raise your final grade by 2%
  - » Constructive participation on class list may raise your grade by 1%

Score	Grade		Score	Grade
100	A		79	B-
99	A		78	C+
98	A		77	C+
97	A		76	C+
96	A		75	C
95	A		74	C
94	A		73	C
93	A		72	C
92	A		71	C-
91	A-		70	C-
90	A-		69	C-
89	A-		68	D+
88	B+		67	D+
87	B+		66	D+
86	B+		65	D
85	B		64	D
84	B		63	D
83	B		62	D
82	B		61	D-
81	B-		60	D-
80	B-		59	D-
			58	F

Policy on Collaboration

- **Assignments/projects/summaries:**
  - » Purpose: familiarization of concepts and details of programming languages
  - » Work on project independently:
    - – No direct sharing of code
    - – No line-by-line assistant
    - – No exchange of code
  - » You are encouraged to ask questions of one another, and to respond to other student's questions (and especially on the email list)
- **Exams:**
  - » Closed-book. No outside assistance is permitted. No additional materials may be used.
  - » No make-up tests unless absence is due to serious illness. Doctor's diagnostic note is required. The final grade will be scaled accordingly.

Paper Presentations

- **1-2 presentations will be expected, needs to be in power point.**
- **We will assign presentations next week.**
  - » Caveat: If someone signs up for a paper and then later drops, we will need to shift the last scheduled person to the empty slot(s) (other volunteers are welcomed and will be solicited in class).
- **Format:**
  - » A mini-conference
  - » Audience will also be given an evaluation sheet to fill out.
  - » 2 Session-Chairs (with prepared questions part of presentation grades).

Paper Presentations

- **Turn in:**
  - » Presenter:
    - – Turn in .pdf of slides
    - – 1 summary
  - » Session Chairs:
    - – Turn in questions & answers
    - – 1 summary
  - » Rest of class:
    - – 1 summary

Project Summaries

1. **What is the problem that the authors are trying to solve?**
   - » Why is the problem important?
2. **What is their approach and how is it original and innovative? (original - compare it against contemporary approaches).**
3. **How is the approach evaluated?**
   - » What are the simplifying assumptions?
   - » What are the strength and weaknesses of their solution?
4. **What are the results/impact of paper**
   - » Why is this paper important?
   - » Did they solve the problem?
   - » Does it have an impact - is it still relevant? Why is it worth reading.
5. **What constructive criticism can you give to the presenter (e.g., would should have been included/excluded, make sure to address 'concepts' covered in the paper and relate how they were covered by the presenter).**
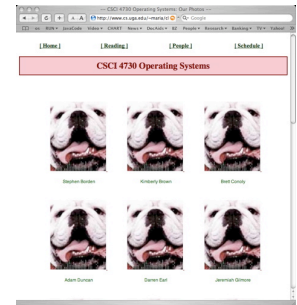
Tentative/past projects for class

- **Lexer & Parser.**
- **Functional Language Project (SML - toy)**
- **Functional Languages (ML – industrial).**
- **Scripting Language (Python).**
- **Logical Language (Prolog).**

Homework 1

- **See schedule for details…**
- **Digital Image -- How to get out of the dog pound (and improve your grade).**

Schedule of Topics

See Web Page

Please check web page often

Subscribe to email list (when set-up)

99 Bottles of Beer in 877 different programming languages (1994)

```
#include <stdio.h>  /* C version */

int main(void)
{
int b;

for( b = 99; b >= 0; b-- ) {
    switch (b) {
        case 0:
            printf("No more bottles of beer on the wall, no more bottles of beer.\n");
            printf("Go to the store and buy some more, 99 bottles of beer on the wall.\n");
            break;
        case 1:
            printf("1 bottle of beer on the wall, 1 bottle of beer.\n");
            printf("Take one down and pass it around, no more bottles of beer on the wall\n");
            break;
        default:
            printf("%d bottles of beer on the wall, %d bottles of beer.\n", b, b);
            printf("Take one down and pass it around, %d %s of beer on the wall.\n"
            ,b - 1 ,((b - 1) > 1)? "bottles" : "bottle");
            break;
        }
}
return 0;
}
```

```
10 REM BASIC Version of 99 Bottles of beer
20 FOR X=100 TO 1 STEP -1
30 PRINT X;"Bottle(s) of beer on the wall,";X;"bottle(s) of beer"
40 PRINT "Take one down and pass it around,"
50 PRINT X-1;"bottle(s) of beer on the wall"
60 NEXT
```

99 Bottles of Beer in 877 different programming languages (1994)

```
;;; Tim Goodwin (tim@pipex.net)    Scheme
(define bottles
    (lambda (n)
        (cond ((= n 0) (display "No more bottles"))
              ((= n 1) (display "One bottle"))
              (else (display n) (display " bottles")))
        (display " of beer")))

(define beer
    (lambda (n)
        (if (> n 0)
            (begin
                (bottles n) (display " on the wall") (newline)
                (bottles n) (newline)
                (display "Take one down, pass it around") (newline)
                (bottles (- n 1)) (display " on the wall") (newline)
                (newline)
                (beer (- n 1))))))
(beer 99)
```

99 Bottles of Beer in 877 different programming languages (1994)

```
#!/usr/local/bin/python
#   python version of 99 bottles of beer, compact edition
#   by Fredrik Lundh (fredrik_lundh@ivab.se)

def bottle(n):
    try:
        return { 0: "no more bottles",
                1: "1 bottle"} [n] + " of beer"
    except KeyError: return "%d bottles of beer" % n

for i in range(99, 0, -1):
    b1, b0 = bottle(i), bottle(i-1)
    print "%(b1)s on the wall, %(b1)s,\n"\
          "take one down, pass it around,\n"
          "%(b0)s on the wall." % locals()
```

```
#!/usr/bin/awk -f
#   awk version of 99 bottles of beer
#   by Whitey (whitey@netcom.com) - 06/05/95

BEGIN {
    for(i = 99; i > 0; i--) {
        print s = bottle(i), "on the wall,", s ",",
        print "take one down, pass it around,"
        print bottle(i - 1), "on the wall."
    }
}

function bottle(n) {
    return sprintf("%s bottle%s of beer", n ? n : "no
more", n - 1 ? "s" : "")
}
```

99 Bottles of Beer in 877 different programming languages (1994)

```
% 99 bottles of beer.    Prolog
% Remko Troncon <spike@kotnet.org>

bottles :-
    bottles(99).

bottles(1) :-
    write('1 bottle of beer on the wall, 1 bottle of beer,'), nl,
    write('Take one down, and pass it around,'), nl,
    write('Now they are alle gone.'), nl.
bottles(X) :-
    X > 1,
    write(X), write(' bottles of beer on the wall,'), nl,
    write(X), write(' bottles
    write('Take one down and p
    NX is X - 1,
    write(NX), write(' bottles
    bottles(NX).
```

```
"Programmer: patrick m. ryan - Smalltalk
 pryan@access.digex.net"http://www.access.digex.net/~pryan

99 to: 1 by: -1 do: [ :i |
    i print. ' bottles of beer on the wall, ' print.
    i print. ' bottles of beer. ' print.
    'take one down, pass it around, ' print.
    (i-1) print. ' bottles of beer on the wall, ' print.
    ]
```

Maria Hybinette, UGA

Introductions

- **Name, major, year?**
- **What are you hoping to learn from the class?**
- **What type of projects are you interested in?**
- **What do you want to do when you graduate?**

Maria Hybinette, UGA

20