Fall 2004

CS 480/680: Comparative Languages

Michael T. Cox
Wright State University - Main Campus

Follow this and additional works at: http://corescholar.libraries.wright.edu/cecs_syllabi

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation
http://corescholar.libraries.wright.edu/cecs_syllabi/120

This Syllabus is brought to you for free and open access by the College of Engineering and Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu.
CS 480-680 COMPARATIVE LANGUAGES

Instructor:
Dr. Michael T. Cox
cst80fac@cs.wright.edu
http://www.cs.wright.edu/~mcox/

Office:
RC343 (phone 775-5126)

Office Hours:
TBA

Prerequisites:
Data Structures CS 400/600.
Experience with programming in imperative languages such as C/C++, Pascal, or Ada.

Meeting Times:
M W 4:10-5:25 PM, Room 303 Oelman Hall
Credit Hours:
4 (3 hr. lecture, 2 hr. lab)

Syllabus

Slides - Access from Syllabus

Old Midterm

Course Description:

"Comparative Languages" is a graduate/undergraduate level introductory course in programming languages. We will cover several basic topics ranging from syntax (BNF) and semantics formalisms (attribute grammars), to data types, scope and extent, type checking, parameter passing methods, expression parsing and other fundamentals of programming languages and language development. The intent of the course is to provide a background in the concepts and constructs of languages, rather than simply providing just a survey of various computer languages. Nonetheless in this class, we will learn and program in three very different languages: Pascal (an imperative language), Java (an object-oriented language), and LISP (a functional language). I expect classes to consist of active discussions in addition to lectures; thus, class participation will be important and encouraged. Because of this reason, and because lectures will contain information not covered in the text and for which the student will be responsible on tests, attendance is crucial. Class lectures and discussions will be complemented by programming assignments, reading assignments, and problem-solving exercises.

We will study concepts using three programming languages.

1. Sun Pascal 4.2 (available from your university account, i.e., paladin)
2. Java (available from your class account)
3. Allegro Common Lisp 6.1 (available from your class account)

Objectives:

The main objectives of the course are
1. to provide a basic survey of language classes
2. to develop an understanding of the basic concepts, constructs and tradeoffs involved in language implementation
3. to ground the theory of languages in existing programming languages
4. to enable you learn to learn new languages

Required Text:


Supplementary Required Readings

Pascal, JAVA, and LISP Resources:

- Sun WorkShop Compiler Pascal 4.2 AnswerBook
- JAVA Tutorial
- LISP FAQ (Frequently Asked Questions)
- Interactive LISP Tutorial
- Common Unix Commands
  
  http://www.wright.edu/cats/docs/docroom/unix/unixcom.htm

Homework

Concerning Academic Dishonesty:

Academic dishonesty will not be tolerated in class, on homework, or during examinations. For a list of examples of cheating see Section X in the Code of Student Conduct in the online Wright State University Student Handbook.

Student Evaluation:
There will be a midterm exam constituting 25% of the grade. The exam will be thought-provoking, as opposed to being an exercise in memorization and recall. You will be tested on your understanding of the principles and your ability to apply them to new and different problems. The final exam will be written likewise, constituting 35% of the course grade. Homework assignments will provide 35% and occasional pop quizzes will be worth 5%. Class participation and attendance will also be significant biasing factors in the evaluation.

A positive bias factor comes into effect when the student is on the dividing line between grades. The bias can put the student over the line to a higher grade. Also important to the calculation of positive grading bias is class “brownie points.” These points are earned in class when a student discovers the instructor making a mistake. The first student that announces the error during lecture receives the point. Student are responsible to keeping tally of their brownie points and reporting them to the instructor at the end of the course (they may also brag about their current tally to the class as each new mistake is discovered). NOTE, all professors make mistakes. Never believe a professor without a critical eye (this is key to the scientific method); instead, challenge them publically.-)

All of the assignments will be submitted in class with hardcopy on the dates specified. A copy will also be submitted electronically before class. If your assignment is incomplete, turn it in and we will grade it for partial credit. If your assignment is late, I will impose a penalty of 15% for one day maximum.

All assignments must be done independently or in groups when instructed. You are encouraged to talk to the instructor, if you encounter any difficulties. When working independently, you may discuss the problem, and the issues that arise, with other students, as long as you clearly indicate who you discussed the problem with. You may not share or discuss your code. In other words, talk about the problem, not about the program itself. Likewise when working in groups, adhere to the same conduct between groups. These are supposed to be common sense guidelines; please talk to the instructor, if you are unsure about what is permissible. If you use algorithms or methods described in books or papers, make sure you include a comment in your program with the appropriate references.

Some assignments will contain one or more extra credit problems. These are designed for students who need extra credit to remedy a poor score and for students who wish to explore particular topics in greater depth. The final grades will be assigned on a curve based on regular credit problems, after which students with extra credit will be scaled up appropriately. This ensures that extra credit can
only help those students who earn it, but not hurt those who don't.

**College Computing Resources:**

Pascal, and JAVA are available on departmental Unix machines (e.g., gamma). Allegro Common Lisp is available only from your class account on cheops.cs.wright.edu.

**Help and Communication:**

College computer labs, software availability, and the CaTS help desk are listed on-line. The CaTS phone number is 775-4827. Many problems such as how to use your university web server can be answered by them.

Most communication regarding the course will be done electronically. Please read your electronic mail (e-mail) regularly (i.e., once or twice per day); the `pine` command, Netscape mail, or others can be used to do this. Announcements will be posted on the newsgroup `wright.cs.cs680-01` which you should also read regularly; the `rn` command can be used to do this, as well as the `gnus` command within `emacs` and with the Netscape newsreader. The newsgroup is also useful for discussing among yourselves various issues pertaining to class (but not solutions to assignments).

Communicate with the instructor by sending e-mail to `cs680fac@cs.wright.edu`. It is important that you use your class account to do this, not your university account or other accounts such as yahoo. All homework assignments MUST be sent from your class account; credit may not be given otherwise. You are encouraged to send any questions about the assignments to the `wright.cs.cs680-01` newsgroup so that everyone can benefit from the questions and answers.

Correspondence: cs680fac@cs.wright.edu