

CS 3180 – Comparative Programming Languages

Course Overview

This course explores the concepts, methods, and data organization underlying modern programming languages. In this course we will explore several different programming styles, including **functional** and **object-oriented** programming. We will look at several languages “under the hood”, exploring how they organize and access data, dispatch method and function calls, and implement higher-order language features. We will use these tools to explore “metaprogramming” – using high-level programming constructs to produce minimal and elegant code.

Learning Outcomes

After completing this course, students will be able to:

1. Demonstrate knowledge of concepts, constructs, and paradigms underlying programming languages.
2. Develop programs in a functional language, and an object-oriented language.
3. Be able to define and employ programming language concepts such as declarative programming, and meta-programming.
4. Be able to apply recursion, list processing, higher-order features, and pattern matching for problem solving.
5. Be able to explain and apply inheritance, polymorphism, dynamic dispatch, strong typing, data abstraction, and modularity in programming.

Course Details

Instructor: Tarek Seleem

✉ – tarek.seleem@wright.edu

Office hours: 472 Joshi @ 11:00-12:00 Tuesdays and Thursdays or by appointment

See the course web page on Pilot for course lectures and assignments.

Equipment needed: Students will need a computer capable of running interpreters and development tools for all languages taught in the course. There are interpreters and development environments available for **Racket**, **Ruby**, and **Python** that run on Microsoft Windows, Apple OS X, and most Linux distributions.

Textbooks: D. Thomas and A. Hunt, *Programming Ruby* (<https://ruby-doc.org/docs/ruby-docbundle/ProgrammingRuby/index.html>) (Electronic) [TH]

D. Sitaram, *Teach Yourself Scheme in Fixnum Days* (<https://ds26gte.github.io/tyscheme/>) (Electronic) [DS]

The Python Tutorial (<https://docs.python.org/3/tutorial/>) (Electronic) [PT]

P. Perrotta, *Metaprogramming Ruby*, Pragmatic Bookshelf, 2010. ISBN-13: 978-1-934356-47-0 (Optional) [PP]

Course Web Page

The course web page is at <http://pilot.wright.edu> (<http://pilot.wright.edu>). Log in using your CaTS username (w000xxx) and password. *It is your responsibility to check this page often for announcements, assignments, and other important information.*

Tentative Lecture Schedule

Week(s)	Topic(s)	Reading
1	Overview, compiled and interpreted languages	Slides
2	Language specification	Slides
3	Functional languages & Racket	DS 1, 2, 3
4	Racket parameters, variables, & scope	DS 4, 5, 6
5	Racket macros, structures, & I/O	DS 7, 8, 9
6	Object-oriented languages & Ruby	TH 1, 2
7	Containers, blocks, iterators	TH 3, 4
8	Inheritance, mixins, regular expressions	TH 5, 6

9	Exceptions, I/O	TH 7, 8
10	Metaprogramming in Ruby	PP 1
11	Introduction to Python	PT 1, 2, 3, 4
12	Object-oriented Python	PT 9
13	Containers, iterators, and operator overloading in Python	PT 6, 7, 8, 9
14	Numeric processing and data science in Python	PT 10

Grading

Take-home Exams (2): 40%

Programming Projects (7): 60%

Homework: For practice only, not graded

Final grades will be assigned according to the university standard grading scheme: 90-100% = A, 80-89% = B, 70-79% = C, 60-69% = D, 0-59% = F. Depending on the overall distribution of final grades, the boundaries *may* be adjusted by the course instructor to ensure consistency from term to term.

Course Policies

Coding Standards

It is your responsibility to read, understand, and adhere to the coding standards explained in the *Coding Standards* document available in the Content -> Handouts section of our Pilot web page. Your ability to develop professional code that conforms to industry-accepted coding standards will be a significant part of your course grade.

Late Assignments

Lab (programming) assignments are due by 11:59 pm on the due date. Late lab assignments will be accepted, but **10%** of the total available points will be deducted for each day late. Labs are considered one day late after 11:59 pm on the due date. At 11:59 pm of each successive day (including weekends) the lab is considered an additional day late until turned in. ***No points will be awarded for labs turned in more than 3 days late.***

Academic Integrity

Discussion of course contents with other students is an important part of the academic process and is encouraged. However, it is expected that course programming assignments, homework assignments, and other course assignments will be completed *on an individual basis*. Students may discuss general concepts with one another, but may not, under any circumstances, work together on the actual implementation of any course assignment. If you work with other students on “general concepts” be certain to *acknowledge the collaboration* and its extent in the assignment. Unacknowledged collaboration will be considered dishonest. “Code sharing” (including code from previous semesters) is strictly disallowed. “Copying” or significant collaboration on any graded assignments will be considered a violation of the university guidelines for academic honesty.

If the same work is turned in by two or more students, *all parties involved will be held equally accountable for violation of academic integrity*. You are responsible for ensuring that other students do not have access to your work: do not give another student access to your account, do not leave printouts in the recycling bin, pick up your printouts promptly, do not leave your workstation unattended, etc. If you suspect that your work has been compromised notify me immediately.

Failure to attend the first day of class, during which time I will explain these academic honesty policies in detail, does not excuse you from following these policies. If you have any questions about collaboration or any other issues related to academic integrity, please see me immediately for clarification.

In addition to the policy stated in this syllabus, students are expected to comply with the Wright State University Code of Student Conduct (<http://www.wright.edu/students/judicial/conduct.html>) and in particular the portions pertaining to Academic Integrity (<http://www.wright.edu/students/judicial/integrity.html>) at all times.

Additional Needs

Students with disabilities or any additional needs are encouraged to set up an appointment with me at their convenience to discuss any classroom accommodations that may be necessary.