Programming Language Concepts

CSCI-344
Term 20145

Handout 1
January 26, 2015

Syllabus

General Information

Instructor: Matthew Fluet
E-mail: mtf@cs.rit.edu
Office hours: Mo 1:00pm – 2:00pm; GOL-3555
Tu 11:00am – 12:00pm; GOL-3555
We 10:00am – 11:00am; GOL-3555
Th 11:00am – 12:00pm; GOL-3555
or by appointment

Lectures: Section 01 MWF 11:00am – 11:50am; GOL-3560

Website: http://www.cs.rit.edu/~mtf/teaching/20145/plc
http://mycourses.rit.edu

Course Description

This course is a study of the syntax and semantics of a diverse set of high-level programming languages. The languages chosen are compared and contrasted in order to demonstrate general principles of programming language design and implementation. The course emphasizes the concepts underpinning modern languages rather than the mastery of particular language details. Programming projects will be required.

Prerequisites

- CSCI-243 (The Mechanics of Programming) and MATH-190 (Discrete Mathematics for Computing)
- or SWEN-250 (Personal Software Engineering) and MATH-190 (Discrete Mathematics for Computing)
- or IGME-309 (Data Structures and Algorithms for Games and Simulations II) and MATH-190 (Discrete Mathematics for Computing)
- or permission of instructor
Course Goals

Although students who have had the introductory sequence have a solid background in basic algorithmic and object-oriented techniques, their perspective is narrow and their understanding of programming languages uncritical. The main goal of this course is to widen students’ perspectives by learning the power of alternative paradigms. Students become more critical and self-aware as software designers and thereby are better able to choose the appropriate paradigm and language for the problem at hand. The second goal is to expose the students to the decisions, both in design and in implementation, that go into developing new programming languages.

Course and Program Outcomes

Course learning outcomes:

- Students will be able to learn new programming paradigms and languages on their own.
  Program outcome(s): 1
  Evaluation: homework, projects, presentations, or exams

- Students will be able to describe programming language syntax formally and semantics informally.
  Program outcome(s): 1
  Evaluation: homework, projects, or exams

- Students will be able to identify appropriate programming languages to use to address the specific needs of a stated problem.
  Program outcome(s): 1
  Evaluation: homework or exams

- Students will be able to explain and apply basic constructs and concepts used in common programming languages.
  Program outcome(s): 1
  Evaluation: homework, projects, or exams

Program Outcomes:

- (CS Undergraduate Program Outcome 1) Demonstrate fluency in high-level programming languages, environments, and tools for computing.
Grades

Grades will be assigned based on the following grading scheme:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance &amp; Participation (lectures)</td>
<td>7.5%</td>
</tr>
<tr>
<td>Attendance &amp; Participation (recitations)</td>
<td>7.5%</td>
</tr>
<tr>
<td>Reading Quizzes (∼12)</td>
<td>10%</td>
</tr>
<tr>
<td>Textbook Feedback</td>
<td>+10%</td>
</tr>
<tr>
<td>Programming Assignments (∼8)</td>
<td>60%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>15%</td>
</tr>
</tbody>
</table>

Final letter grades will be assigned based on the following grading scale:

<table>
<thead>
<tr>
<th>Letter grade</th>
<th>Numeric grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>[93, ∞)</td>
</tr>
<tr>
<td>A</td>
<td>[90, 93)</td>
</tr>
<tr>
<td>B+</td>
<td>[87, 90)</td>
</tr>
<tr>
<td>B</td>
<td>[83, 87)</td>
</tr>
<tr>
<td>B−</td>
<td>[80, 83)</td>
</tr>
<tr>
<td>C+</td>
<td>[77, 80)</td>
</tr>
<tr>
<td>C</td>
<td>[73, 77)</td>
</tr>
<tr>
<td>C−</td>
<td>[70, 73)</td>
</tr>
<tr>
<td>D+</td>
<td>[65, 70)</td>
</tr>
<tr>
<td>D</td>
<td>[60, 65)</td>
</tr>
<tr>
<td>F</td>
<td>[0, 60)</td>
</tr>
</tbody>
</table>

Important Dates

May 18 (Mon.): Final Exam (2:45pm – 4:45pm; GOL-3560)

Text Books

Required:
Title: Programming Languages: Build, Prove, Compare (draft)
Author: Norman Ramsey (with Samuel L. Kamin)
Note: Available for purchase at CS Dept. front office (∼$35, cash only)

Course Policies

Attendance & Participation

Students are required to attend and expected to participate in class. Participation means being an engaged student, not simply attending class. During lectures, asking and answering questions is the primary means of participation; during recitations, contributing to group problem solving and presenting solutions are the primary means of participation.
The use of cell phones and audio players is prohibited during class. If you must take a phone call, please leave the classroom immediately and do not return until you have ended the phone call.

The use of a laptop (or notebook or netbook) computer is permitted during class only for the purpose of taking notes. Persistent use of a laptop for other activities will result in 0 credit for your Attendance & Participation grade.

Assigned readings should be completed before the lecture section. You are responsible for the material in assigned readings, whether covered during lecture or not.

**Final Exam**

There will be a final exam; see above for the date. The final will be comprehensive and will cover material from the entire course, including readings, lectures, and assignments.

The final exam must be taken at its scheduled time. Any exam conflicts must be reported to the instructor by the end of Week 10 (see the RIT Final Examination Policies).

**Late Policy**

Assignments that are submitted electronically (most assignments) will generally be due at 11:59PM on the due date. Assignments that are submitted in person will generally be due at the beginning of a class period.

Assignments are to be submitted on time. However, to accommodate the occasional difficulty with meeting an assignment due date, each student begins the term with five “extension tokens.” By spending an extension token, you will receive a 24-hour extension on a single programming assignment. To spend an extension token, you must e-mail the instructor before the assignment is due; you cannot spend an extension token after an assignment’s due date has passed. You may spend at most one extension token on a single assignment. After spending five extension tokens, late assignments will not be accepted.

**Regrading**

After a graded exam or assignment has been returned, you have one week to bring any questions about grading to the instructor’s attention. No grade adjustments will be made after this time.

**Academic Integrity**

As with all courses, the RIT Honor Code, RIT Academic Honesty Policy apply. See the Department of Computer Science’s statement on academic integrity for more details.

In this course, all submitted work must be your own work (i.e., written or programmed by you alone, unless explicitly stated otherwise) and must include acknowledgments
of any collaborators or sources (other than course text books or handouts) used to produce your submission.

You are encouraged to discuss course material with other students. Discussion of assignments is also allowed, but sharing solutions or code is not allowed.

**Common Course Policies**

See the Department of Computer Science’s [Common Course Policies](#) for more details about rescheduling an exam, course withdrawal, disability services, and academic integrity.

**Disclaimer**

I reserve the right to make any changes to the syllabus as I deem necessary throughout the course. Minor changes, such as assignment due dates, will be announced orally during class and posted on the course mailing list and home page. Major changes, such as grading percentages, will additionally be provided in writing.

**Acknowledgements**

Portions of this course material based upon similar courses offered at Tufts University (Norman Ramsey, Samuel Guyer).