Course Description

This course is an introduction to the formal study of programming languages, demonstrating important intellectual tools for the precise description of programming languages and investigating the essential features of programming languages using these tools. Topics include: dynamic semantics (such as operational semantics); static semantics (such as type systems); proofs by induction on structures and derivations; formal treatment of essential programming-language features (such as assignment, scope, functions, objects, and threads). Both written and programming assignments will be required.

Enrollment Requirements

- Prerequisites: (CSCI-603 and CSCI-605 and CSCI-661) with grades of B or better or ((CSCI-262 or CSCI-263) and CSCI-344).

Course Goals

The precise description of the semantics of programming languages is required to thoroughly understand the meaning of computer programs.

This course studies the formal semantics of programming languages. Students will learn about important intellectual tools such as operational semantics and type systems and will investigate essential features of programming languages using these tools. While the focus is on formal models of small languages, the applicability of these formal models to “real” programming languages will be demonstrated. Students will gain an appreciation of the design decisions
(and design mistakes) in extant programming languages and will be prepared to study the programming-languages research literature.

This course does not cover tools and techniques for describing the concrete syntax of programming languages (e.g., scanners and parsers); such topics are covered in CSCI-741 Compiler Construction.

Topics

- Abstract syntax
- Judgements and inference rules
- Operational semantics
- Lambda calculus
- Simply-typed lambda calculus
- Type-safety proof (preservation, progress, substitution)
- Simply-typed lambda calculus extensions
- Subtyping
- Parametric polymorphism, recursive types, and existential types
- Concurrency with shared-memory and with message-passing
- Advanced concepts in object-oriented programming

Note: The order in which topics are discussed in lectures will likely differ from that given above. Furthermore, not all topics will receive equal (or, possibly, any) time. Exams will only cover topics explicitly discussed in lecture or in an assigned reading.

Course and Program Outcomes

Course learning outcomes:

- Students will be able to give precise definitions of programming-language features using operational semantics and type systems.
  Program outcome(s): 2
  Evaluation: exams, homework assignments

- Students will be able to translate programming-language specifications from mathematical notation to code.
  Program outcome(s): 2
  Evaluation: exams, homework assignments

- Students will be able to prove properties of inductively defined sets (e.g., well-typed programs).
  Program outcome(s): 2
  Evaluation: exams, homework assignments

- Students will be able to effectively make use of the research literature in programming languages.
  Program outcome(s): 2, 4
  Evaluation: exams, homework assignments
Program Outcomes:

- (CS Graduate Program Outcome 2) Demonstrate a depth of knowledge in a selected area in the discipline.
- (CS Graduate Program Outcome 4) Pursue professional positions or further graduate studies.

Grades

Grades will be assigned based on the following grading scheme:

<table>
<thead>
<tr>
<th>Participation:</th>
<th>5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework Assignments:</td>
<td>50%</td>
</tr>
<tr>
<td>Technical Perspective:</td>
<td>10%</td>
</tr>
<tr>
<td>Mid-term Exam 1:</td>
<td>10%</td>
</tr>
<tr>
<td>Mid-term Exam 2:</td>
<td>10%</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>(60, 70)</td>
</tr>
<tr>
<td>F</td>
<td>[0, 60)</td>
</tr>
</tbody>
</table>

Important Dates

- **September 21 (Mon.):** Mid-term Exam 1
- **October 21 (Wed.):** Mid-term Exam 2
- **December 7 (Mon.):** Final Exam
Text Books

Suggested:

Title: Types and Programming Languages
Author: Benjamin C. Pierce
Publisher: The MIT Press
ISBN: 978-0262162098
Website: https://www.cis.upenn.edu/~bcpierce/tapl/
RIT Library e-book: https://albert.rit.edu/record=b3882573-S3

Title: Practical Foundations for Programming Languages (2nd Edition)
Author: Robert Harper
Publisher: Cambridge University Press
ISBN: 978-1107150300
Website: https://www.cs.cmu.edu/~rwh/pfpl.html
PDF Draft Copy: https://www.cs.cmu.edu/~rwh/pfpl/2nded.pdf

Additional:

Title: Advanced Topics in Types and Programming Languages
Editor: Benjamin C. Pierce
Publisher: The MIT Press
ISBN: 978-262162289
Website: https://www.cis.upenn.edu/~bcpierce/attapl/
RIT Library e-book: https://albert.rit.edu/record=b3780803-S3

Title: The Formal Semantics of Programming Languages
Author: Glynn Winskel
Publisher: The MIT Press
ISBN: 978-0262731034

Course Policies

Attendance & Participation

Students are required to attend (either in person or via Zoom) and expected to participate in the course. Participation means being an engaged student: asking and answering questions (in person, via Zoom, on Discussions), not simply attending class.

The use of cell phones and audio players is prohibited during in-person class meetings. 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 in-person class meetings only for the purpose of taking notes. Persistent use of a laptop for other activities will result in 0 credit for your Participation grade.

Note: This course is being offered with both the Online Flex and Full Flex options. A student can choose to complete all course requirements online either for the entire semester (Online Flex) or as needed throughout the semester (Full Flex). The primary expectation is that flex students will attend class meetings synchronously using Zoom.
and participate via Zoom and on Discussions (with a slight emphasis on Discussions, depending on the ease of participating via Zoom). Contact the instructor (at any time during the semester), if you wish to select one of these options.

Students who are required to quarantine or isolate should contact the instructor to select Full Flex for the duration of their absence. Again, the primary expectation is that flex students will attend class meeting synchronously using Zoom and participate via Zoom on Discussions. For circumstances that go beyond standard quarantine or isolation, students should contact an appropriate administrative staff member (e.g., the Disability Services Office (DSO), assistant or associate dean for student success or academic services, or academic advisor) and ask that they contact the instructor, who will make special arrangements suited to the situation.

**Mid-term Exam**

There will be two mid-term exams; see above for the dates.

The mid-term exams must be taken at their scheduled times (modulo the natural flexibility afforded by a take home / online exam). Make-up mid-term exams will not be administered, unless exceptional circumstances have been discussed with the instructor in advance of the exam date and/or other arrangements have been made.

**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 lectures and assignments.

The final exam must be taken at its scheduled time (modulo the natural flexibility afforded by a take home / online exam). Any exam conflicts must be reported to the instructor by the end of Week 10 (see the RIT Final Examination Policies).

**Late Policy**

Assignments are to be submitted on time. However, to accommodate the occasional (ordinary) difficulty with meeting an assignment due date, each student begins the term with three “extension tokens.” Using an extension token grants a 24-hour extension on a single homework assignment (no extensions for technical perspective assignments). An extension token is automatically applied to “Late Submissions” on myCourses (work submitted after the “Due Date” but before the “End Date”); an extension token cannot be used after the assignment’s “End Date” has passed. Only one extension per assignment will be granted. After using three extension tokens, late submissions will not be accepted. The instructor will not discuss or answer questions about a homework assignment after the assignment’s “Due Date” has passed.

For extraordinary difficulties, contact an appropriate administrative staff member (e.g., the Dissability Services Office (DSO), assistant or associate dean for student success or academic services, or academic advisor) and ask that they contact the instructor, who will make special arrangements suited to the situation.

**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 and the Student Academic Integrity Policy apply. See the Department of Computer Science’s statement on Student 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.

Technical Perspective

Each student will write a “Technical Perspective” for a recent programming-languages research paper. A “Technical Perspective” is a two page paper that explains the importance of the research and the specific contributions of the paper; it is written by an expert for non-expert computer scientists and helps to place the paper within a larger research context.

You will need to choose a research paper (a list of suitable papers will be provided), read and understand the paper (which, in turn, will require reading additional background material), and write a “Technical Perspective.” Be aware that writing a short “Technical Perspective” is more challenging than writing a long paper summary.

For more information, see tech_perspec.pdf/tech_perspec.html.

COVID-19 Addendum

We are all aware of the unique circumstances of this fall semester resulting from the worldwide COVID-19 SARS-2 pandemic. RIT has consulted federal, state, and local guidelines and policies to implement a safe, yet educational environment for students, faculty and staff. These guidelines, located at https://www.rit.edu/ready/ are routinely updated as conditions change.

What do these mean for this class? When we meet in person everyone will wear a mask that covers their mouth and nose at all times and have freshly washed or sanitized hands. In class, students will sit in assigned seats in the locations designated by faculty. We will not congregate in hallways, bathrooms or classrooms prior to or after class. Any presence of fever or other COVID-19 symptoms will be reported on the RIT Daily Health Screen Monitoring (https://www.rit.edu/news/rit-launches-daily-health-screen-monitoring-covid-19-symptoms); please notify myself (Matthew Fluet, mailto:mtf@cs.rit.edu) so that the best way to accommodate your learning can be planned.

The instruction mode for this course will be Blended A/B with the ABA-S format. The registrar will assign each student to either Group A or Group B for the course. Alternating for each class
meeting day, one group will attend in person and the other group will participate synchronously via Zoom.

Homework assignments must be submitted through myCourses, as either source code (programming components) or PDF (written components). Electronic preparation (e.g., LaTeX) of homework is suggested; handwritten assignments will be accepted (please use a document scanner or a mobile scanning app like Microsoft Lens or Adobe Scan). Mid-term exams and the final exam will be administered take home / online.

This course is being offered with both the Online Flex and Full Flex options. A student can choose to complete all course requirements online either for the entire semester (Online Flex) or as needed throughout the semester (Full Flex). The primary expectation is that flex students will attend class meetings synchronously using Zoom and participate via Zoom and on Discussions. Contact the instructor (at any time during the semester), if you wish to select one of these options.

Students who are required to quarantine or isolate should contact the instructor to select Full Flex for the duration of their absence. Again, the primary expectation is that flex students will attend class meeting synchronously using Zoom and participate via Zoom on Discussions. For circumstances that go beyond standard quarantine or isolation, students should contact an appropriate administrative staff member (e.g., the Disability Services Office (DSO), assistant or associate dean for student success or academic services, or academic advisor) and ask that they contact the instructor, who will make special arrangements suited to the situation.

In the event that RIT moves to fully online instruction for the fall semester (or the instructor is required to quarantine or isolate), class meetings will move to synchronous participation via Zoom for all students.

We will talk in class about these expectations to ensure that we all are comfortable with what is happening during class. I encourage your communication about any special needs or concerns. Together we will learn about Programming Language Theory in a safe and productive format!

Disclaimer

The instructor reserves the right to make any changes to the syllabus deemed 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 University of Washington (Dan Grossman), Cornell University (Andrew Myers, Dexter Kozen, Nate Foster), Indiana University (Amal Ahmed), Harvard University (Greg Morrisett).