BU CAS CS 320 (Fall, 2023)
Concepts of Programming Languages (Part 2)

Syllabus

- **Semester** Fall, 2023
- **Lecture Times**: TR: 11:00-12:15; 12:30-1:45
- **Classroom**: CAS B12; LSE B01
- **Instructor**: Marco Gaboardi
- **Instructor's Office**: CDS 1019
- **Instructor's Office Hours**: Tu-Th 2:15-3:30pm
- **Teaching Fellow**: Qiancheng (Robin) Fu and Zachery Casey
- **Teaching Fellows' Office Hours**: Robin: Th 2:00-3:30pm, Fr 1:00-2:30pm (CDS725) Zach: Tu-Th 3:30-5:00 (CDS725)
- **Teaching Assistants**: Anming Gu and Jason Wang
- **Teaching Assistants' Office Hours**: Jason: Mon: 12:30 - 2:30; Fri: 2:30 - 3:30 (CDS 362) Anming: Mo-We 4:30 - 6 (CDS 362)
- **Final**: A two-hour in-class exam covering both Part 1 and Part 2.
- **Overview**: Concepts of Programming Languages (CPL) is a course that introduces students to some fundamental concepts in programming language design and implementation. The primary goal is to allow students who complete this subject to have a good feel for the elements of style and aesthetics of programming and a good command of the major techniques for controlling complexity in programming. Moreover, students who will complete this subject will have an understanding of how programming languages can be described formally and how they can be implemented.

  OCaml is a functional programming language that makes pervasive use of types in capturing programming invariants. We will be primarily doing programming in OCaml while making ocassionally use of Python so as to demonstrate concretely that the concepts learned in OCaml can be readily applied in Python, one of the most popular language in the world.

  Ultimately, we would like to make a convincing argument that programming can be a great deal of fun if you do it the right way!

  In the second part of the class we will look at formal grammars and operational semantics as formal tools to describe precisely the syntax and semantics of programming languages. We will also have hand on experience on how to implement them.

- **Grades** The final score is calculated using the following formula:

$$\begin{aligned}
\text{Your Score for Part 2} &= 4\%\cdot(\text{homework on grammars}) \\
&+ 4\%\cdot(\text{homework on operational semantics}) \\
&+ 4\%\cdot(\text{quiz01}) \\
&+ 4\%\cdot(\text{quiz02}) \\
&+ 8\%\cdot(\text{Interpreter part 1}) \\
&+ 8\%\cdot(\text{Interpreter part 2}) \\
&+ 8\%\cdot(\text{Interpreter part 3})
\end{aligned}$$

  We will have also a homework0 worth 2% bonus points.

- **Homework Assignments** Homework assignments will be **posted** on the following tentative dates and due the day the next assignment is posted:

  - **Mon Oct 30** Homework0
  - **Mon Nov 6** Homework on Formal grammars.
  - **Mon Nov 13** interpreter part 1 + 1 exercise operational semantics
  - **Mon Nov 27** interpreter part 2 + 1 exercise operational semantics
  - **Mon Dec 4** interpreter part 3

  An assignment that is turned in after its due time is accepted but penalized according to the following policy.

  - 20% point deduction if the assignment is turned in within 24 hours after its due time.
  - 50% point deduction if the assignment is turned in between 24 and 48 hours after its due time.
  - no credit if the assignment is turned in more than 48 hours later after its due time.

  From time to time, students may be requested to explain in details their solutions to the instructor, the TFs, and/or the TAs. Those who cannot adequately explain their solutions may see that their acquired points be deducted partly or wholely.

- **Academic Integrity**: We adhere strictly to the standard BU guidelines for academic integrity. For this course, it is perfectly acceptable for you to discuss the general concepts and principles behind an assignment with other students. However, it is not proper, without prior authorization of the instructor, to arrive at collective solutions. In such a case, each student is expected to develop, write up and hand in an individual solution and, in doing so, gain a sufficient understanding of the problem so as to be able to explain it adequately to the instructor. Under *no* circumstances should a student copy, partly or wholly, the completed solution of another student. If one makes substantial use of certain code that is not written by oneself, then the person must explicitly mention the source of the involved code.