CSC301: Fundamentals of Programming Languages

Syllabus – Fall 2024

Time: Section 1 MWF 10-10:50, Location: Chafee Social Sci Center 273

Webpage: https://lutzhamel.github.io/CSC301 or BrightSpace
Prerequisites: CSC212

Instructor:
Prof. Lutz Hamel
email: lutzhamel@uri.edu
office: Tyler 251

Course Description

Language enables thought. In this course we study a class of formal languages known as programming languages. Like natural languages, these formal languages enable us to reason about algorithms and procedures to solve computational problems on computers. However, their formal nature restricts the kind of meanings particular language constructs can assume and therefore makes them amenable for the execution on a computer.

Over the years many different programming language dialects have evolved to address particular technical issues, e.g., object-oriented languages, real-time languages, database query languages, logic languages, etc. Here we study the major structures of modern programming languages. Understanding not only the syntax of a language but also the semantics and implementation techniques of this language will allow you to design better programs. Having deeper insights into the design of a programming language will also enable you to learn new programming languages much faster. Having a thorough understanding of today's languages allows you to design the programming languages of tomorrow.

Objective

Upon completion of this course

- You will be able to discern and contrast the major programming language paradigms in use today.
- You will be able to pick an appropriate language for the job at hand.
- You will have deeper insight into the evolution of programming languages.

Text

Modern Programming Languages: A Practical Introduction, Adam Brooks Webber, Franklin, Beedle & Associates, Any Edition.

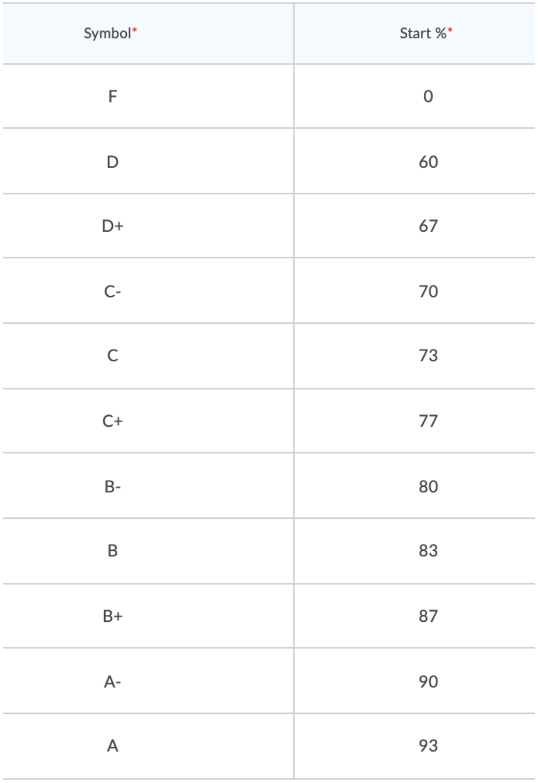
Software

Throughout this course we will be using various programming language and software development environments including Asteroid and Prolog. More details will be given on the course website.

Grading

Assignments, Quizzes	50%
Midterm	25%
Final	25%

Grading Key



Symbol*	Start %*
F	0
D	60
D+	67
C-	70
C	73
C+	77
B-	80
B	83
B+	87
A-	90
A	93

Policies

- Check the website (often)! I will try to keep the website as up-to-date as possible.
- **Promptness, participation,** and **adequate preparation** for each class are expected. If you are absent, it is your responsibility to find out what you missed (e.g. handouts, announcements, assignments, new material, etc.)
- **Make-up quizzes** and **exams** will **not** be given without a valid excuse, such as illness. If you are unable to complete a scheduled examination due to valid reasons, please inform myself, or the department office in Tyler Hall, prior to the exam time. Under such circumstances, you are not to discuss the exam with any other class member until after a make-up exam has been completed.

- All work is to be the result of your own individual efforts unless explicitly stated otherwise. **Plagiarism, unauthorized cooperation or any form of cheating** will be brought to the attention of the Dean for disciplinary action. See the appropriate sections (8.27) of the University Manual. See below for details on programming assignment policies.
- **Software piracy** will be dealt with exactly like stealing of university or departmental property. Any abuse of computer or software equipment will subject to disciplinary action.
- Any student with a documented disability should contact me early in the semester so that we can make reasonable accommodations to support your success in this course. You should also contact Disability Services for Students, Office of Student Life, 330 Memorial Union, 874-2098

Anti-Bias Statement: We respect the rights and dignity of each individual and group. We reject prejudice and intolerance, and we work to understand differences. We believe that equity and inclusion are critical components for campus community members to thrive. If you are a target or a witness of a bias incident, you are encouraged to submit a report to the URI Bias Response Team at www.uri.edu/brt. There you will also find people and resources to help.

***Academic Enhancement Center*:** Located in Roosevelt Hall, the AEC offers free face-to-face and web-based services to undergraduate students seeking academic support. Peer tutoring is available for STEM- related courses by appointment online and in-person. The Writing Center offers peer tutoring focused on supporting undergraduate writers at any stage of a writing assignment. The UCS160 course and academic skills consultations offer students strategies and activities aimed at improving their studying and test-taking skills. Complete details about each of these programs, up-to-date schedules, contact information and self-service study resources are all available on the AEC website, uri.edu/aec.

Programming Assignment Policy

Any programming assignment that is specified as **Individual Work** must be done on your own. While you may discuss general solutions, design ideas and algorithms with classmates, **YOU MAY NOT:**

- **share code with other students**
- **look at any other student's code**
- **use code provided to you by anyone else**
- **use code that you find on the Internet**

If you use code that has been provided for you by the instructor, or that you have used in a prior assignment, include in your comments where the code came from.If you use code that is not your own include in your comments where the code came from. If you ever have a question about what is acceptable when working on a programming assignment, please contact your instructor.

Generative AI tools like ChatGPT are very good at writing code. While these tools can be useful to programmers solving large problems, we want you to learn the fundamentals of programming. Therefore, you are expected to write all code on Individual Work by yourself or use templates provided by the instructor. You may use generative AI tools in a tutorial fashion where you ask general questions about coding and problem solving. But you **MAY NOT** use them to do your assignments. When doing work for this course, **follow these rules** if you are using generative AI tools:

- You may consult the AI tools on general questions
- You **MAY NOT** copy and paste your assignments or code into the tool
- You **MAY NOT** copy and paste any results from the tool into your assignments
- If you use an AI tool while working on an assignment, you **MUST** treat the results as code provided by a third party and document this in your comments providing the prompt that you used and the result of the prompt.
- If you use programming constructs that are not taught in class, you **MUST** cite where you found them and why you have chosen to use them instead of the constructs taught in class.

Any violation of these rules may result in a grade of 0 on the assignment. In addition, you may be reported to the Dean and the Office of Student Life. See the University Manual for more information about the potential consequences of cheating. https://web.uri.edu/manual/chapter-8/chapter-8-2/.

Tentative Schedule

* Why study programming languages?
* Meet our languages: Asteroid and Prolog
* Asteroid – The Imperative Basics
* Types and Type Systems
* Exploring more of Asteroid's Types
* Functional Programming with Asteroid
* Recursion
* Memory management
* Polymorphism
* Logic Programming with Prolog
* Language specification & Implementation
* Formal semantics