

# **CSC301: Fundamentals of Programming Languages**

Syllabus – Fall 2022

**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](mailto: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	40%
Midterm	20%
Final	20%
Attendance	20%

Class attendance is mandatory. You will be given one unexcused absence per semester otherwise an unexcused absence will count against your attendance grade.

## 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.
- **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 is welcome to contact me as early in the semester as possible so that we may arrange reasonable accommodations. As part of this process, please be in touch with Disability Services for Students Office at 302 Memorial Union, Phone 401-874-2098.

***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](http://www.uri.edu/brt). There you will also find people and resources to help.

***Disability Services for Students Statement:*** Your access in this course is important. Please send me your Disability Services for Students (DSS) accommodation letter early in the semester so that we have adequate time to discuss and arrange your approved academic accommodations. If you have not yet established services through DSS, please contact them to engage in a confidential conversation about the process for requesting reasonable accommodations in the classroom. DSS can be reached by calling: 401-874-2098, visiting: [web.uri.edu/disability](http://web.uri.edu/disability), or emailing: [dss@etal.uri.edu](mailto:dss@etal.uri.edu). We are available to meet with students enrolled in Kingston as well as Providence courses.

***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](http://uri.edu/aec).

## ***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