

Fall 2020

CS 635-101: Computer Programming Language

Wallace Rutkowski

Follow this and additional works at: <https://digitalcommons.njit.edu/cs-syllabi>

Recommended Citation

Rutkowski, Wallace, "CS 635-101: Computer Programming Language" (2020). *Computer Science Syllabi*. 133.

<https://digitalcommons.njit.edu/cs-syllabi/133>

This Syllabus is brought to you for free and open access by the NJIT Syllabi at Digital Commons @ NJIT. It has been accepted for inclusion in Computer Science Syllabi by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

CIS 635

Concepts of Programming Languages

Textbook: Not required. All material will be posted on-line on canvas.

Instructor: Wallace Rutkowski
email: wallace.rutkowski@njit.edu
office: GITC 4413
phone: 973-596-5483

Grading: Programming projects and other assignments will be posted on canvas

Programs	30%
Assignments	30%
Final Project	30%
Attendance/Participation	10%

Schedule of Topics:

1 Review: Formal Grammars

2 Parsing, LL and LR languages

Top down, recursive descent

Bottom up shift-reduce parsers

Attribute grammars, synthesized/inherited attributes

CFG, BNF, EBNF

CFG examples for common language features of C++, Java, Python etc.

3 Binding times, Scope, Referencing environment

Storage classes, Non-local references

Comparison of languages that do/don't allow nested function definitions
ie Java vs Javascript

Function parameter passing methods

Parameter passing in modern languages

Java, C#, C++, Python, Javascript etc.

Stack implementation of functions,

Activation Record Instances (ARI), Recursion

4 Arrays. Arrays implementation in C++ , Java, Python

Pointers and arrays in C, C++, dynamic array allocation using pointers

C++ pointers vs Java references

Order of operations in arithmetic statements, Side effects

Short circuit boolean evaluation,

5 Functions as parameters:

Referencing environment, deep/shallow/ad-hoc binding

Implementation of deep binding using static chain/display

Functions as parameters comparison for modern languages

Non-local references, Resolution of non-local references

using static chain pointers and displays

- Implementation of dynamic scope:
 - Shallow access/Deep Access/Central table
- 6 Computer Architecture and its Relation to Compilers and Language Design
 - Memory Hierarchy
 - Data Representation
 - Instruction set architecture
 - Register allocation
 - Parallelism
 - Pipelining
 - Multi-core
- 7 Midterm Examination
- 8 Object Oriented Programming (OOP),
 - Abstract Data Types (ADT), classes
 - Public/Private members, Data/Method members,
 - Class (static)/Instance members
- 10 Implementation of virtual methods, vtables
 - Non-local references in OOP, inner classes
- 11 Container classes, C++ multiple inheritance
 - Problems with multiple inheritance, Java Interfaces
- 12 Advanced issues in OOP:
 - Why Derived * can be converted to Base * but cannot convert Derived ** to Base **
 - Why Derived * to Base * conversion in C++ makes C++ arrays/pointers unsafe in OOP
 - Problems with dynamic allocation/de-allocation in code that can throw exceptions and solutions
- 13 Concurrency, Semaphores, Monitors
 - Multithreading, Java monitors, synchronized methods
- 14 Event driven/multithreading programming
 - Inner classes for event handling
 - GUI construction and thread safe programming
- 15 Final Examination

The Provost has asked that we include the following statement:

“Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at: <http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf>.

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. Any student found in violation of the code by cheating, plagiarizing or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university. If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu”