SYLLABUS
Old Dominion University • College of Sciences • Department of Computer Science
CS 355 • Principles of Programming Languages • Fall 2020

Instructor
Dr. Andrey Chernikov
• Email: achernik@cs.odu.edu, please include “CS355” in the subject line
• Office location: ECS 3311A, phone: 757-683-7732
• Office hours (both physical and virtual): Fridays 2:00pm to 6:00pm
• Email for appointments outside office hours

Course Information
Meeting hours  Mondays, Wednesdays, and Fridays 12:00pm to 12:50pm

Mode of delivery  Scheduled In-Class Meetings, Batten Arts-Letters 1012: a face-to-face section in a traditional classroom, adjusted for all COVID-19 university safety protocols

Catalog description  Survey of significant features of programming languages. Language types including imperative, functional, logical, and object-oriented are covered. Concepts include lexical and syntactic analysis, type systems, flow control, modularity, and parallel programming. Small programs in several languages required. Laboratory work required.

Prerequisites
• CS 252 Introduction to Unix for Programmers
  An introduction to Unix with emphasis on the skills necessary to be a productive programmer in Unix, Linux, and related environments. Topics include command line shells, files and directories, editing, compiling and common command line utilities.
• CS 250 Problem Solving and Programming II (a grade of C or better)
  Design issues arising in software systems and C++ programming techniques aiding in their solution. Topics include the software life cycle, methods of functional decomposition, design documentation, abstract data types and classes, common data structures, dynamic data structures, algorithmic patterns, and testing and debugging techniques.

Course Objectives  Students will learn the elements that make up programming languages, the rationale behind language design choices, and the way in which different language elements interact. Upon completing this course, they should be able to
• learn new programming languages on their own by utilizing a variety of sources,
• access and comprehend the documentation for a programming language,
• be able to choose a suitable language to a particular project, and
• understand the trends in the development of programming languages.

Topics to Be Covered
• History, Translation Methods
• Names/Bindings/Scope, Data Types
• Abstract Data Types, Object-Oriented Design
• Assignment/Expressions, Control Structures, Subprograms, Exceptions, Concurrency
• Functional and Logic Programming Languages

Required textbook  Sebesta, Concepts of Programming Languages, 12th edition, Pearson, 2019

Current course information, additional materials, and assignments:
https://www.blackboard.odu.edu (check frequently)
Attendance policy  Class attendance is not factored into the grade. However, attending class meetings is strongly encouraged as they provide a dedicated environment to stay focused and to ask questions. Students are responsible for all material covered and announcements made in class.

Disability  Students are encouraged to self-disclose disabilities that have been verified by the Office of Educational Accessibility by providing Accommodation Letters to their instructors early in the semester in order to start receiving accommodations. Accommodations will not be made until the Accommodation Letters are provided to instructors each semester.

Assignments

• 80% of the overall grade is allocated to bi-weekly programming projects that include the application of concepts learned and interpretation of the results
• 20% is allocated to the final take-home exam that will focus on the concepts learned throughout the course and include minor programming examples

Grades  The final percentage score will be computed as follows:

$$\text{final score} = \frac{\text{individual total project score}}{\text{maximum total project score}} \times 85 + \frac{\text{individual final exam score}}{\text{maximum final exam score}} \times 20,$$

The extra 5% is intended to offset minor unfortunate circumstances, including, but not limited to, health issues, memory lapses, and equipment malfunctions. Such circumstances should not be brought to instructor’s attention. The final letter grade will be looked up from the following table.

<table>
<thead>
<tr>
<th>final score</th>
<th>letter grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 45 50 55 60 65 70 75 80 85 90 95 105</td>
<td>F D- D D+ C- C C+ B- B B+ A- A</td>
</tr>
</tbody>
</table>

Submission rules

• All assignments will be published and submitted through the Blackboards system. No other medium will be accepted without prior instructor’s permission. All files included in a submission should be collected into a single package, preferably zip.
• All submitted code is expected to compile and run on the CS Linux cluster with command line tools only. No extra time will be allowed for porting from another environment.
• Each late day after the project submission deadline will be penalized by a 10% grade reduction. For example, a project that is 4 days late will receive no more than 60% of allocated points. Midnight is considered the beginning of a new day.
• Make-up assignments will be allowed only if justified by a documented evidence (e.g., doctor’s note) of student’s inability to complete the corresponding work before the original deadline. The final exam can be taken outside the scheduled interval only with the permission from the dean’s office.
• An interruption in the internet connection will not be recognized as a reason for deadline extension, unless accompanied by a note from the university IT department. Therefore, it is strongly advised to complete the assignments well before the deadlines.

Academic integrity

• All assignments must be completed individually. No person, except the instructor or the teaching assistant, can be asked for help with solving the assignments. This requirement does not apply to setting up the computer and common software.
• The internet and any printed materials can be consulted as needed. However, all sources used in the graded work must be cited. All submitted code must be written by the student.
• Students are cautioned against replicating problem solutions found on the internet, as they are sometimes incorrect and often poorly written. The instructor may question the student to determine if submitted work corresponds to student’s proficiency and assign the grade accordingly.
• Students must be familiar with and abide by the University Honor Code: https://www.odu.edu/about/monarchcitizenship/student-conduct/code.