

Course Assessment Document for CS 333 Programming Languages

Departmental Outcomes

1. Proficiency in computational thinking
2. Ability to analyze systems at the three levels of computer science: theory, software, and hardware
3. Proficiency in the design and implementation of algorithms using multiple programming languages
4. Ability to apply computational thinking to a diverse set of problems and disciplines
5. Ability to communicate effectively and collaborate with others
6. Ability to adapt to new challenges and computational environments

Course Description

This course is a survey of programming languages and paradigms. We will focus on the design of programming languages and compare and contrast different language families including imperative, object-oriented, functional, and logic paradigms. Topics include syntax, context-free grammars, parsing, semantics, abstract representations of programming processes and structures, memory management, and exceptions. Students will undertake small programming projects in various languages and more extensive projects in two languages of their choice. Students will present the characteristics of their chosen languages to their peers at the end of the term.

Prerequisites: CS 231

Rationale for prerequisites: Students require sufficient experience with programming and computational thinking to understand and analyze the design of programming languages. The prerequisite guarantees they have had two CS courses and used two different languages.

Desired Course Outcomes

- A. Students demonstrate an understanding of different language paradigms and implement algorithms in each paradigm.
- B. Students demonstrate an ability to independently learn programming languages.
- C. Students demonstrate an ability to describe the syntax, semantics and functionality of different languages in a common, rigorous manner.
- D. Students demonstrate an understanding of the relationship between language and design.
- E. Students work with partners to learn one or more languages and present them to the class.
- F. Students present algorithms, languages, and their characteristics in an organized and competently written manner.

We will disseminate the desired course outcomes to students via the course web page, syllabus and in class.

Course Matrix

Outcome	Activities	Method of Assessment	Departmental Outcome
A	Lectures, projects, homeworks	Exams, project reports, homework	1, 2, 3, 6
B	Lectures, projects, homeworks	Exams, project reports, homework	1, 3, 6
C	Lectures, projects, homeworks	Exams, project reports, homework	1, 2, 3
D	Lectures, projects, homeworks	Exams, project reports, homework	1, 2, 3, 4
E	Projects	Project reports and presentations	5
F	Projects, homeworks	Project reports, homeworks	5

Grade Calibration Matrix

Outcome	Meaning of the grade A
A	The student implements all required algorithms in each paradigm and explores other algorithms or languages on their own initiative.
B	The student finds and makes use of a variety of resources, including the professor, to learn two or more new languages and demonstrates the ability to use the language by both describing the features and using them in example programs.
C	The student describes the syntax, semantics, and functionality of multiple languages in at least two different forms (e.g. mathematical and functional). The descriptions are clear, well-documented, and may include functional code that would form part of an interpreter for the language.
D	The student demonstrates a clear understanding of the features of a language that support different design styles and can coherently address the issues that arise when selecting an appropriate language given a task and the design of the solution.
E	Work is shared evenly between the partners. The students are able to effectively divide the task into parts, and each group member completes their task on time and is able to integrate it into the project. Partners communicate effectively and work collaboratively. The language presentations are clear, divided evenly between the students, and provide a strong introduction to the language.
F	Reports are well-written, concise, and clear. The reports clearly describe the language characteristics, example code, and language design issues. The reports are written so that students outside the course could understand them. The language pages are appropriate as tutorials on the language.

Outcome	Meaning of the grade B
A	The student implements all required algorithms in each paradigm.
B	The student finds and makes use of appropriate resources to learn two new languages and demonstrates the ability to use the language by both describing the features and using them in example programs.
C	The student describes the syntax, semantics, and functionality of two languages in at least one common form (e.g. mathematical or functional). The descriptions are clear and follow the conventions provided in the textbook or lecture.
D	The student demonstrates an understanding of the features of a language that support different design styles and can identify the major issues that arise when selecting an appropriate language given a task and the design of the solution.
E	Work is shared between the partners. The students are able to divide the task into parts, and each group member completes their task on time and is able to integrate it into the project. Partners communicate sufficiently to integrate their work. The language presentations are clear, divided fairly event between partners, and cover the required material.
F	Reports are well written and clear. The reports describe the required language characteristics and design issues and make use of example code to support the text. The reports are written so that students taking the course could understand the work. The language pages could serve as tutorials on the language with further revision.

Outcome	Meaning of the grade C
A	The student implements most of the required algorithms in each paradigm.
B	The student finds and makes use of a few resources to learn two new languages and demonstrates some ability to use the language by describing a small set of features and creating working variations of some found example programs.
C	The student describes the syntax, semantics, and functionality of at least one language in a mathematical or functional form. The descriptions may be inconsistent but generally follow the conventions provided in the textbook or lecture.
D	The student demonstrates an understanding of the most significant features of a language that support different design styles and can identify some issues that arise when selecting an appropriate language given a task and the design of the solution.
E	Work is unevenly shared between the partners, possibly duplicated. The students divide the work based on which student is likely to finish the work. Not all group members complete their work on time. Partners communicate, but may have difficulty integrating their work. The language presentations are not well organized and do not cover all of the required material.
F	Reports describe the work, but may take the form of a narrative of the student's difficulties rather than focusing on the language characteristics. The reports describe some of the required language characteristics and have some example code. The reports assume the reader is familiar with the language and are not appropriate as a tutorial.

Outcome	Meaning of the grade D
A	The student implements some of the required algorithms in each paradigm.
B	The student has difficulty finding and using language resources, may attempt to learn only one language, and has difficulty writing functional example programs in the chosen languages.
C	The student describes the syntax, semantics, and functionality of at least one language in a non-rigorous manner. The descriptions may be inconsistent or approximate the conventions provided in the textbook or lecture.
D	The student demonstrates some ability to differentiate the features of various languages that support different design styles and has difficulty identifying issues that arise when selecting an appropriate language given a task and the design of the solution.
E	The student works on their own or is loosely affiliated with their partner. Partners do not communicate often or effectively and have difficulty integrating code. The language presentations are not organized, do not cover all of the required material, and do not demonstrate any depth of understanding.
F	The student's reports are incomplete or not well written. The report includes descriptions of some language characteristics, but little explanation. Example code is copied, not well documented, or not well-suited as an example. The language pages contain little information about meaning of statements and do not sufficiently describe important language features.

A student who receives an F does not meet the criteria for a D or any higher grade.