# Programming Languages

## CS333, Fall 2011

| Instructor        | Dr. Brian S. Eastwood (bseastwo at colby.edu)  
|                  | Roberts 224C  
|                  | 859-5852  
| Lectures         | Monday, Wednesday, Friday  
|                  | 9:00 am – 9:50 am, Roberts 221  
| Office hours     | Monday 10:00 am – 12:00 pm  
|                  | Thursday 1:00 pm – 3:00 pm  
|                  | Or by appointment  
| Text             | Allen Tucker and Robert Noonan  

## 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.

## Objectives

1. Students demonstrate an understanding of different language paradigms and can implement algorithms in each paradigm.
2. Students demonstrate an ability to independently learn programming languages.
3. Students demonstrate an ability to describe the syntax, semantics, and functionality of different languages in a common, rigorous manner.
4. Students demonstrate an understanding of the relationship between language and design.
5. Students work with partners to learn two or more languages and present them to the class.
6. Students present algorithms, languages, and their characteristics in an organized and competently written manner.

## Prerequisites

CS231, Data Structures and Algorithms. Students are assumed to have a strong foundation in computational thinking, and should have deep experience with at least one programming language.
**Grading**

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>20</td>
</tr>
<tr>
<td>Projects</td>
<td>30</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>15</td>
</tr>
<tr>
<td>Final Exam</td>
<td>15</td>
</tr>
<tr>
<td>Presentation</td>
<td>15</td>
</tr>
<tr>
<td>Class Participation</td>
<td>5</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

**Homework.** Approximately weekly homework assignments will consist of problem sets, writing, and short programming exercises. Homework is due by class time on the given due date. Because these assignments will typically be discussed in class, late homework is not accepted. You are encouraged to discuss the homework concepts with your classmates, but the work you submit must be your own.

**Projects.** Several (7 – 8) more substantial programming projects will involve writing programs that demonstrate programming language concepts. The grading policy on projects is that the tasks specified explicitly in the project description will constitute about 85% of the assignment. If you complete the specified parts of the project properly, and produce a high-quality writeup, it's worth up to a B+ grade. In addition, the project description will include a variety of extensions, or you can come up with your own. Completing one or more extensions—along with high quality work on the remainder of the project—will earn you some flavor of A.

Each assignment has a written component in which you will reflect on the problem you are solving and the effectiveness of your solution. This should be completed on your personal wiki page or a personal web site (you are free to choose), and a link to this page should be sent to the instructor. *Do not post complete code on your website.* The code for assignments should be submitted to the instructor via email (use a single zip file). The write up should include a description of the problem, a discussion of any required theory, a presentation of the results, and a discussion including possible enhancements.

For full credit, assignments should be submitted by 11:59:59 pm on the due date. Late assignments will incur a 5% penalty per calendar day up to a maximum of 8 days.

**Exams.** There will be one in-class midterm exam and a cumulative final exam. The midterm exam is tentatively scheduled for **Wednesday, 19 October 2011**. The final exam period for this class is **Thursday, 15 December 2011 at 6:00 pm**.

**Presentation.** Throughout the course, you will be working in pairs to learn two programming languages that are initially unfamiliar to you. Each group will build online learning resources about these programming languages in the form of a blog or other internet website. Towards the end of the course, students will complete oral presentations on the characteristics of their chosen languages. The completed online learning resources and the oral presentations represent a significant effort in the course, roughly equivalent to a midterm exam.

**Collaboration.** You may work in pairs on programming projects if you wish. When completing assignments in pairs, each team member's grade will be based on the grade earned by the project, but individual grades may be raised or lowered depending on each team member's contribution to the project. Include in the project write up a description of the contribution from each team member. Also
note that projects completed by groups are expected to be more extensive than those completed by individuals.

Collaboration and discussion on projects is encouraged. However, you need to understand everything your code (or your group's code) does, and any code you turn in must be your own (or your group's). Good scholarship requires that all collaboration must be acknowledged. If you collaborate on the solution of a problem set or project, I expect that you list your collaborators. Additionally, whenever you use a source in any way, that usage must be documented in the assignment turned in. Sources include books, articles, magazines, the Internet, and communication with other people.

**Participation.** Active participation in the classroom is a critical component of education, and this depends on each student arriving at class prepared and on time. You are expected to attend all classes, and class participation forms a component of your grade. Extreme cases of absenteeism may result in failure of the course. You are expected to have completed any reading that will be discussed in the upcoming class and you are responsible for all information covered in class, whether or not you are there.

**Text**


This book will be available at the Colby bookstore or online (e.g., http://www.amazon.com/Programming-Languages-Allen-Tucker/dp/0072866098).

**Schedule**

A tentative schedule is available on the course website; this will be updated to include reading assignments.