Computer Science 331  
Principles of Programming Languages  
Spring 2018  
Dr. Steven K. Andrianoff

Time and Place:  
MWF 11:30 – 12:20 Walsh 204

Instructor:  
Dr. Andrianoff (andrianoff@sbu.edu)  
Walsh 113  
375-2053  
Office hours:  
Mon 2:00 – 3:30 p.m.  
Tues 1:30 – 2:30 p.m.  
Wed 2:00 – 3:00 p.m.  
Thur 2:00 – 3:30 p.m.  
Fri 10:30 – 11:30 p.m.  
(Other times by appointment)

Course web page:  
http://www.cs.sbu.edu/andrianoff/CS331/index.htm  
In general, announcements, readings, and assignments for CS 331 will be given in class AND published on the course web page. Students are expected to check that page regularly for news, and are nonetheless responsible for any assignment announced in either manner.

Text:  

Catalog description:  
A study of the fundamental principles around which programming languages are designed. Topics include language specifications, syntax and semantics, data types, control statements, support for abstraction, and program organization. Students will be exposed to issues involving compilation, interpretation, lexical analysis, and parsing as well as to languages in each of the major paradigms. The course includes three lecture hours per week. 3 credits.

Prerequisites:  
CS 234 – Programming Methodologies

Course objectives:  

1) To understand the issues and underlying principles in the design of a programming language. To understand, as well, the implementation issues of programming language features.

2) To be familiar with formalisms used to describe the syntax and semantics of a programming language.

3) To be exposed to programming languages other than Java for the purpose of comparison of features. In particular, students will study Scheme (as a functional programming language) in more depth as an example of a language in a contrasting paradigm.
Tentative Schedule:

Basically, we will be covering the entire text, although we will be covering some portions fairly lightly. The pace of the course will be determined somewhat by the abilities of the students, but an approximate notion of where we will be is given below. Specifics will be given on the course web page as the semester progresses.

<table>
<thead>
<tr>
<th>Chapters</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2 - The groundwork</td>
<td>1 week</td>
</tr>
<tr>
<td>3-4 - Formal descriptions of languages</td>
<td>2 weeks</td>
</tr>
<tr>
<td>5 - Names and Scopes</td>
<td>1 week</td>
</tr>
<tr>
<td>6-8 - Standard Programming Constructs: Data &amp; Control</td>
<td>2 weeks</td>
</tr>
<tr>
<td>9-10 - Subprograms</td>
<td>2 weeks</td>
</tr>
<tr>
<td>11-12 - Object-orientation</td>
<td>2 weeks</td>
</tr>
<tr>
<td>13-14 - &quot;Advanced&quot; features</td>
<td>2 weeks</td>
</tr>
<tr>
<td>15 - Functional Programming (Scheme)</td>
<td>1 week</td>
</tr>
<tr>
<td>16 - Logic Programming</td>
<td>1 week</td>
</tr>
</tbody>
</table>

Course Policies:

Grades:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework, papers, programming assignments</td>
<td>40%</td>
</tr>
<tr>
<td>Quizzes (best 5 of 8)</td>
<td>30%</td>
</tr>
<tr>
<td>Final exam (comprehensive)</td>
<td>30%</td>
</tr>
</tbody>
</table>

Homework:

Daily written homework problems will be assigned, collected, and graded. Note that written homework assignments are an important component of the course. Although this course does not have a laboratory component, you will be given several programming assignments to complete that illustrate concepts from the course.

Papers:

You will be required to write several (2-4) papers during the course of the semester. You will be required to research a programming language and report on your findings. You will also write a paper on the Scheme programming language.

Quizzes: Approximately eight quizzes will be given. Only the five highest quiz scores will be kept. There are generally no make-ups for missed quizzes.

Final exam: The final exam is comprehensive and is scheduled for Friday, May 4, at 1:10 p.m.
Late policy:

If homework is received the day that it was due, but after the deadline, the penalty will be 10%. An additional 10% penalty will be assessed for each day after the work is due up to the point where the graded work is returned. Late work will not be accepted once the graded work is returned.

The policy for papers is slightly different. A late paper will be subject to 10% penalty for each day it is late up to a maximum of 50%.

Attendance policy:

There is no attendance requirement, however students are expected to attend all of the classes and will be responsible for all assignments. Attendance will be monitored. More than three absences is considered excessive.

Absences from class, excused or not, do not exempt the student from completing the assigned work. If you miss a class it is your responsibility to arrange with your instructor for the work to be made up in a timely manner.

Classroom Etiquette

Use of cell phones or other personal electronic devices during class is rude and inconsiderate of others. Please turn off cell phones and similar devices during classes and labs. This policy does not apply to personal laptops which may be used for notes and lab exercises.

Academic integrity policy:

Academic dishonesty is inconsistent with the moral character expected of students in a university committed to the spiritual and intellectual growth of the whole person. It also subverts the academic process by distorting all measurements. It is a serious matter and will be dealt with accordingly. A list of unacceptable practices, penalties to be assigned, and procedures to be followed in prosecuting cases of alleged academic dishonesty may be found in the Student Handbook.

Students are expected to read and abide by the department’s Academic Practices and Policies, a copy of which will be distributed with the course syllabus. Unless other instructions are explicitly stated all graded work will be subject to the policy

"Individual Project With Limited Collaboration: In particular, you may receive help from the following persons, in addition to an instructor in this course: any St. Bonaventure University student enrolled in CS 331, and any other person specifically approved by your instructor. You may use the following materials produced by other students: NONE."

In addition, if you do collaborate with anyone other than the instructor, there must be a note to that effect at the top of the solution you turn in.

Academic dishonesty in any form will not be tolerated. Typically the first offense will result in a zero on the assignment. Repeated offenses will likely result in a failing grade for the course. Any offense deemed punishable will also be referred to the Dean of Arts and Sciences.

Services for Students with Disabilities:

Students with disabilities who believe that they may need accommodations in this class are encouraged to contact the Disability Support Services Office, Plassmann 100D, at 375-2065 as soon as possible to better ensure that such accommodations are implemented in a timely fashion.
Computer Science Goals and Learning Objectives

This course has the specific goal of ensuring that students understand the unifying and differentiating properties of various programming languages and to ensure that each computer science major has engaged in a major writing assignment as part of the major curriculum. Two of the language papers (Scheme and "free choice") are designed to support these goals.

Within the context of the Department's formal goals and objectives, the course content and format directly address Objectives 1 and 2 of Goal #1. The Scheme project directly addresses Objective 4 of Goal #2. The two papers reflect work towards the first part of Objective 4 of Goal #3. These goals and objectives are reproduced below:

Goal #1: Discipline Specific Learning

Students will be able to understand and apply the theoretical tools of computer science to standard problems from the field.

Objectives:

1. Students will learn core concepts of the discipline as determined by a nationally recognized professional computer science education organization.
2. Students will understand and analyze algorithms written in pseudo-code.

Goal #2: Reasoning and Inquiry Skills

Students will be able to read, write, and analyze program fragments and complete programs.

Objectives:

4. Students will complete at least one project using a significant technology in the discipline without direct instruction.

Goal #3: Communication Skills

Objectives:

4. Students will be given the opportunity to present results of their work in oral and written forms; this will include the presentation of posters and/or papers intramurally and extramurally.