Course Syllabus

CS 131 - Computer Science I

Dr. Steven K. Andrianoff
Fall, 2015

Time and Place:

<table>
<thead>
<tr>
<th>Lecture:</th>
<th>MWF 10:30 – 11:20 AM</th>
<th>Walsh 204</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab:</td>
<td>M 2:30 – 4:30 PM</td>
<td>Walsh 101</td>
</tr>
<tr>
<td></td>
<td>(Lab Instructor: Dr. David Levine, <a href="mailto:dlevine@sbu.edu">dlevine@sbu.edu</a>)</td>
<td></td>
</tr>
</tbody>
</table>

Instructor:

Dr. Andrianoff (andrianoff@sbu.edu)
Walsh 113
Office hours: 1:30 – 2:30 Mon
1:30 – 2:30 Tues
2:30 – 4:00 Wed
1:30 – 3:00 Thur
11:30 – 12:30 Fri
Other times by appointment
(These hours may be changed during the first couple of weeks of class)

Course web page:

In general, announcements, readings, assignments, and laboratory exercises for CS 131 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.

Texts: Roman Lysecky and Adrian Lizarraga, Programming in Java

This is an online textbook that is available at zyante.com. Instructions for acquiring text:
1. Sign up at zybooks.com
2. Enter zyBook code: SBUCS131Fall2015
3. Subscribe using any credit card (the cost is $48 and is valid until Jan 3, 2016)

Catalog Description

Computer Science I is the first course in the computer science major sequence. The course introduces the object-oriented approach to software design using the programming language Java. Topics covered are software design, implementation and testing, basic computer organization and source code translation. No previous programming experience is presupposed. The course consists of three lecture hours and one two-hour laboratory per week. 4 credits. Fall & Spring.

Note: The course may not be taken for credit without the laboratory component.

Expected Outcome

Students will be able to design programs using the object-oriented design methodology and will be able to implement and test programs using the object-oriented programming language Java. They will be introduced to the principles of software engineering, in particular to the practice of designing large programs using modules, which can be specified, implemented, and tested independently. Students will be able to manipulate the simple and composite data types. They will also be introduced to basic computer organization and data representation at the machine level.

Course Outline

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introductory Material (computer architecture, data representation, programming languages, object-oriented programming) (Chapter 1)</td>
<td>3 weeks</td>
</tr>
<tr>
<td>Java Programming Fundamentals (Chapter 2)</td>
<td>3 weeks</td>
</tr>
<tr>
<td>Introduction to Classes, Objects, &amp; Methods (Chapters 6, 7)</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Flow of Control (selection and repetition statements) (Chapters 3, 4)</td>
<td>3 weeks</td>
</tr>
<tr>
<td>Arrays and ArrayLists (Chapters 5, 7)</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Recursion (Chapter 11)</td>
<td>1 week</td>
</tr>
</tbody>
</table>
Course Grade

The final grade is determined by three components: quizzes, a final examination, and other written assignments (including laboratory assignments, programming assignments, and written homework assignments). The three components are weighted as follows:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes</td>
<td>30%</td>
</tr>
<tr>
<td>Final Examination</td>
<td>30%</td>
</tr>
<tr>
<td>Assignments</td>
<td>40%</td>
</tr>
</tbody>
</table>

Note: A passing grade (minimum of 60%) must be earned from the quizzes and final examination for a student to receive a passing grade for the course.

Quizzes

Quizzes will be twenty-minute, closed-book exams held during class. Approximately eight quizzes will be given, however, only the five best quiz grades will contribute to the final grade, the others will be dropped. There will be no make-ups for unexcused absences when quizzes are given.

Final Examination

The final examination will be a 2-hour closed book examination. It is scheduled for Tuesday, May 15, at 8:00 a.m. Every student is required to take the final examination. The final examination is comprehensive.

Laboratories and Programming Assignments

There is a two-hour lab each week. Most labs require a write-up that must be done using a word processor and have a cover sheet. The penalty for a late lab write-up is 10% per day late. Lab write-ups will be due Monday at class time following the week assigned. **Attendance at laboratory sessions is mandatory: each absence over 1 will result in a reduction in grade by one half letter grade (i.e. one letter grade modifier) for the course.** Any material from a missed lab must be handed in on the date due and is subject to the late penalty.

Programming assignments are assignments that are completed individually by each student outside of formal lab. The late penalty for programming assignments is the same as for labs – 10% per day late. Please note that both labs and programming assignments are governed by the department’s *Academic Practices and Policies* distributed with this syllabus.

Homework Assignments

Homework assignments will be taken from the textbook as well as in the form of handouts. The online assignments will be corrected and recorded electronically. The homework must be completed on time to earn full credit. No late assignments are accepted after the graded work is returned.
Attendance

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

Attendance at laboratory sessions is mandatory. You are allowed one absence from lab. Absences beyond the first one will result in the deduction of one half letter grade (one grade modifier) from the final grade for the course.

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 Honesty Policy

The Computer Science Department takes academic honesty violations (e.g., plagiarism) seriously. The department issues a pamphlet, Academic Practices and Policies, that outlines our expectations in this area. All assignments, including work done both inside and outside of the lab, are governed by this document. Inappropriate conduct and the penalties for it are defined in this document. A copy is distributed to all computer science students each semester. Please review it: Ignorance of the policies and procedures is not an excuse for violating them.

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.
Computer Science Goals and Learning Objectives

Computer Science I helps students establish a base upon which to build towards the longer-term goals of the program. The course addresses the following goals and objectives that have been established by the department:

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.
5. Students will identify the various components of a computer-based system and explain how they integrate to form a coherent solution. Students will document the components in a style commonly used by software professionals.

Goal 2: Reasoning and Inquiry Skills

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

Objectives:

2. Students will enhance existing (larger) programs to add capabilities and/or improve the quality of code.
3. Students will design test suites for and run debugging sessions on programs they have written as well as on programs written by others.

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, Doyle Room 26, at 375-2065 as soon as possible to better ensure that such accommodations are implemented in a timely fashion.