Syllabus
CIS 3020: Advanced Programming Fundamentals
Fall 2008

Course Description

CIS 3020: Advanced Programming Fundamentals

Credits: 3; Prerequisites: MAC 2311 or MAC 3472, programming experience

A fast paced introduction to computer science for students with prior programming experience. Major concepts of computer science and the process of computer programming including object-oriented programming, procedural and data abstraction and program modularity.

This course uses the Java programming language. The course discusses and concentrates on the following major topics: (1) object-oriented programming, (2) procedural abstraction, (3) data abstraction, and (4) program modularity and state information. Java is our tool for discussing these topics.

Students often refer to this course as “the Java course” which it is NOT (note italics in previous paragraph). The intent of this course is to cover the major concepts of computer science and the process of computer programming. Java is merely being used as a vehicle for talking about programming and our coverage of Java technology will be incomplete. You will learn this language since part of the process of being a computer scientist or engineer is understanding a programming language so you can use it to solve problems. But, recognize that this course’s emphasis is much broader, encompassing an overview of the entire field of computer science.

Students are hereby advised that this is a challenging course intended for computer science majors — regardless of your actual major, you are expected to approach the course as if you were a CS major. Prior programming experience is required! If you do not have prior programming experience, then drop this course and take CIS 3022: Programming Fundamentals for CIS Majors 1 and CIS 3023: Programming Fundamentals for CIS Majors 2 or CGS 2414: Programming with Java (which is geared for non-majors wanting to learn a little about programming).

Course Coordinator

Douglas D. Dankel II
Office: E330 CSE
Phone: 392-1387
E-mail: ddd@cise.ufl.edu

Dr. Dankel is responsible for the lectures, examinations, and the coordination of the activities of the Teaching Assistant.

Teaching Assistants

TBA
E-mail: ???@cise.ufl.edu
Office Hours: tba

The Teaching Assistant is in charge of the laboratory sessions and assisting with any questions that you have concerning Java and the assignments.
Required Text

Highly recommended: Program Development in Java: Abstraction, Specification, and Object-Oriented Design, Barbara Liskov with John Guttag, Addison Wesley, 2001. This text is highly recommended for the course and will be the text for all reading assignments. This should be available in the textbook stores in town or on-line.

Recommended: Java in a Nutshell (5th Edition), David Flanagan, O'Reilly Press, 2005. This should be available in the textbook stores in town or on-line.

Recommended: An Introduction to Programming and Object Oriented Design Using Java (2nd Edition), Jaime Nino and Frederick A. Hosch, John Wiley, New York, 2005. This should be available in all of the textbook stores in town or on-line.

Copies of the overhead slides used in class will be available in PDF format on the course e-Learning System site. Students should print out their own copy of these to use for taking notes in class so you can focus on what is said in lecture rather than hurriedly trying to write everything down during class. Note that material has been left out of the notes – this is material for you to fill in during lecture. I will not provide the completed set of notes. If you miss class, see another student for the materials you missed.

Attendance

Since roll call will not be taken in lecture, your attendance is generally optional. However, since the students who do the best in this course are those who attend the course’s lectures, your attendance is strongly encouraged and expected! To do your best in this class, you should consider your attendance mandatory! Please be prompt in arriving for class and do not start packing up your materials before the class is over. Turn off all cell phones prior to entering the classroom. If a cell phone goes off during lecture I will have a pop quiz!

Periodic pop quizzes will be given during the lecture and discussion sessions at various times throughout the semester. There will be no makeup pop quizzes for any pop quiz that you might miss. No excused absences will be given. Missing a quiz will result in a grade of zero. The lowest two pop quiz grades will be dropped when computing your final grade in the course.

Similarly, an unexcused absence from an examination will result in a grade of zero for that examination. Makeup examinations will not be given. If you have a valid excuse for missing an examination (see below), the grade of “EX” (for “excused”) will be recorded. Your other examination grades will be used to extrapolate your excused grade.

The only valid excuses for missing an examination are prior written approval from the instructor or a documented medical emergency.

Cooperative Learning

During the semester, we will be using a technique known as cooperative learning. Using this technique you will, in a structured manner, learn from each other. The majority of the assigned homeworks will be solved using groups of 2 or 3 individuals, with a single solution submitted for the group. Everyone’s grade in the group will be based on the quality of the submitted solution and verbal answers given by a randomly selected member of the group. Studies have shown that using this technique benefits everyone in the group.
Grading

Your grade will be based on one of two scales:

<table>
<thead>
<tr>
<th>OP1</th>
<th>OP2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15%</td>
<td>10%</td>
<td>Examination 1</td>
</tr>
<tr>
<td>20%</td>
<td>15%</td>
<td>Examination 2</td>
</tr>
<tr>
<td>30%</td>
<td>15%</td>
<td>Examination 3</td>
</tr>
<tr>
<td>-</td>
<td>25%</td>
<td>Final Examination</td>
</tr>
<tr>
<td>20%</td>
<td>20%</td>
<td>Pop Quizzes, in-class exercises, and individual homework</td>
</tr>
<tr>
<td>15%</td>
<td>15%</td>
<td>Group programming assignments</td>
</tr>
</tbody>
</table>

Dr. Dankel reserves the right to vary each of these percentages by no more than 5%.

Examinations 1-3 will occur during regular class periods throughout the semester. We will post final course grades using OP1 in the grade book on e-Learning System as soon as all of the course materials are graded. We will also identify what is the highest grade you can achieve should you take the final examination (OP2) and make a perfect score, a 90, an 80, and a 70. All students must identify by midnight on Monday December 15th if they plan to take the final. The final is scheduled for Wednesday, December 17th from 7:30 a.m. – 9:30 a.m.

Once the total percentage (TP) is computed, a student’s letter grades will be assigned as shown in the table below. Note that grades are not given – students earn their grades! The class will not be graded on a curve.

<table>
<thead>
<tr>
<th>Total Point Percentage Range</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP ≥ 90</td>
<td>A</td>
</tr>
<tr>
<td>90 &gt; TP ≥ 88</td>
<td>B+</td>
</tr>
<tr>
<td>88 &gt; TP ≥ 80</td>
<td>B</td>
</tr>
<tr>
<td>80 &gt; TP ≥ 78</td>
<td>C+</td>
</tr>
<tr>
<td>78 &gt; TP ≥ 70</td>
<td>C</td>
</tr>
<tr>
<td>70 &gt; TP ≥ 68</td>
<td>D+</td>
</tr>
<tr>
<td>68 &gt; TP ≥ 60</td>
<td>D</td>
</tr>
<tr>
<td>60 &gt; TP</td>
<td>E</td>
</tr>
</tbody>
</table>

Cheating and Academic Dishonesty

Unfortunately, it is necessary to mention the subject of cheating. The exercises given in this class require the development of short to medium length programs or program fragments, and there are always many correct solutions to any non-trivial programming problem. It is sometimes difficult for students and instructors to determine what constitutes cheating or academic dishonesty in this setting. Since the goal of this course is to learn the concepts required in the construction of programs, the student must actually engage in the creative process of constructing these programs — simply producing a working program is not enough. Each exercise, if carried out by the student, will give the student understanding, or will reinforce the student’s understanding, of an important computer science concept. Hence, the student working on individual or group assignments is permitted to inspect related problems and ask questions of the TA and others in the class about the problem and related problems, but the student is not permitted to copy the work of any others. This includes but is not limited to the TA, previous TAs, or others (students or non-students) who are taking the class, who previously have taken the class, or who have not taken this course. Likewise, students allowing others to copy their own work are guilty of cheating.
One way to help adjudicate possible cases of forgery is to analyze the level of understanding each student has in their own work. Since this understanding is at the center of the course goals, each student must strive to have a complete understanding of every exercise and the solution they submit.

Individual homework exercises assigned for this course are **not** team projects unless the instructor explicitly tells you so. You may consult with others when attempting to develop your solutions to these individual assignments. It is legitimate for two individuals to **plan** together the general approach that they will **individually** use in solving a problem. But once any **significant** details of the solution starts, collaboration should cease. This does not mean that students may not help each other – there are just limits to “helping”. You should not develop a single solution working together as a team with other students except when you are doing so on the required team exercises. Two identical or nearly identical solutions to the same problem will be regarded as evidence of over-collaboration and will be dealt with as cheating. The borderline, where simply consulting with others becomes working as a team or copying, is a gray area. If you have any doubts, you probably are working too closely and should stop — go off and work by yourself.

Because computer resources are shared, it is essential that all students use them in a way that respects the rights of others. Any attempt to copy other student’s assignments, destroy other’s data or code, or deny the use of the computer to others is unethical and is considered a violation of academic honesty guidelines. The bottom line is, **do not misuse the computer!**

**Do not cheat! Do not copy others work!** Immerse yourself in the class. Learn the material. The benefit and enjoyment you will receive will be much more valuable than any consequences of cheating. **Do not even think about cheating!** You do not realize how easy it is for us to hand-check for those who have cheated and to prove it to others. Note that we have electronic means to detect collaboration, which is even better than our hand-checks, which we will use to check for cheaters! Individuals who have misrepresented work as being their own or who have assisted another will receive as a **minimum**: a grade of **zero** on that assignment **and** a decrease of **one letter grade** on their final course grade. This is **in addition** to any other penalties given by Student Affairs. Every individual in this class should examine the Academic Honesty Guidelines and Student Conduct Code in the **University of Florida Undergraduate Catalog** for more details. Ignorance of these Guidelines is no excuse!

**How to do Well in this Course**

The students who get the most out of this course will be the ones put in the most effort. We will cover a large amount of material in a short time. If you cannot put in the required time to learn the material, you will do poorly. Based on my past experience in teaching this course, you should plan on devoting a **MINIMUM** of ten hours outside of class **PER WEEK** to this course. **DO NOT FALL BEHIND** in the course material of significant difficulties will result.

If you want to do well in the course, attend all lectures, read the assigned readings of the book **before** coming to class, actively participate within your assigned group, and start your assignments **early**. **Starting early** really helps because a solution to some of the problems may not grow in your mind right away when you first look at an assignment. Starting on the problems early and letting your mind work on them in background is often the most effective and efficient way to complete the assignments.

**If you are having difficulty, you owe it to yourself to get help!** Interact with the other members of your group when working on group assignments – bounce ideas off each other. Help each other learn! The TAs and class coordinator are also more than happy to assist you. Do not be afraid to come and see us, but please observe our posted hours. We will hold numerous office and laboratory hours.

You and your group when working on group projects (not the TAs or anyone else) are responsible for solving the assignments within the allotted time limits. The TAs can answer specific questions about the programming assignments, general approaches on how to attack a problem, and your
program solutions. BUT remember: the TAs are NOT there to write your programs or to debug them for you. Please have specific questions ready when you approach them.

We would really love to have everyone in the class earn an “A”. If you work hard and master the material presented, you will learn some powerful and fundamental programming techniques that can and will help you immeasurably when you tackle bigger programming problems later in school and in your career.

**Course Schedule**

The following are important dates for this class this semester:

1. Examination 1: tba - ~October 6 (to be decided by class)
2. Examination 2: tba - ~November 5 (to be decided by class)
3. Examination 3: December 10
4. Final Examination: December 17, 7:30 a.m. – 9:30 a.m.

The dates for the first two examinations will be decided by a class vote. Assignments will be announced during the semester and posted electronically on the class web page. Since the highly recommended text has 15 chapters and the semester is 15 weeks long, one chapter will be highly recommended reading for each week.

**Student with Disabilities**

Students requesting classroom accommodation must first register with the Dean of Students Office. The Dean of Students Office will provide documentation to the student who must then provide this documentation to the course coordinator when requesting accommodation.

**ABET Outcomes**

This course addresses the following ABET outcomes:

(c) an ability to design hardware and software systems, components, or processes to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

(e) an ability to identify, formulate, and solve hardware and software computer engineering problems, accounting for the interaction between hardware and software.

(k) an ability to use the techniques, skills, and modern engineering tools necessary for computer engineering practice.