Instructor Information

Dr. Ben Schafer
316 ITTC Building
schafer@cs.uni.edu
319-273-2187

Office Hours
- MWF, 11:00-11:50 AM, 1:00-1:50 PM
  - While not necessary, to reserve an appointment you can use:
- Anytime my office door is open
- Send me an email and ask for a specific time.

Course Information

Time and Place:
- MWF 2:00-2:50 AM, ITTC 328

Required Text:

This course will not use a required textbook. Instead we will occasionally use publically available readings from the Internet.

Course Description

This course aims to help you develop an understanding of current design methodologies for developing computer programs in an object oriented language / model. That is, working with programs written using collections of independent objects that collaborate to achieve some goal. This course does so by giving you the opportunity to design, critique, and implement OO solutions in Java.

Your skill as a programmer depends largely on how many ways you can think about problems and solutions. At the most abstract level, flexibility allows you to be creative. At a more concrete level, flexibility allows you to learn new kinds of programming languages more effectively. And, at the most concrete level, such flexibility allows you to use individual programming languages more effectively.

Outcomes

By the end of the semester you should have a solid grasp of the object-oriented approach to software development. You should:

1. Know the basic features of OO techniques
2. Know the vocabulary of OO modeling
3. Be able to analyze a real-world situation in an object-oriented way
4. Be able to design an object-oriented model containing multiple classes and collaborations
5. Be able to implement such a model in Java using OO programming techniques.
6. Be able to consider multiple designs for solving the same problem and understand the tradeoffs between these alternate designs.
Class Website

- http://www.cs.uni.edu/~schafer/2530/
  - This is the main website for the course. It will contain lecture notes, assignments, announcements and supplemental class materials.

Course Structure and Grading Policies

The final grade you earn in this course will be based on the points accumulated as described below.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Quantity</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
<td>~15 of varying weight – adjusted to 200 total points</td>
<td>40%</td>
</tr>
<tr>
<td>Exams</td>
<td>2 midterms and a final – each adjusted to 100 points</td>
<td>60%</td>
</tr>
</tbody>
</table>

Assignments

There will be at least two types of assignments this semester

- Programming assignments -- Programming assignments are designed to take what you have learned in lecture and apply these skills to a "complete" program on a larger scale. Please pay attention to the deadlines for every PA. The complexity of these assignments will vary, and some may count for more credit than others. The grading of programming assignments will be based on the quality of your code as well as your adherence to general and assignment specific directions. For more information, see the Programming Assignment Collection Policy.
- Written assignments -- these activities will vary over a wide variety of formats but will most often consist of either written (paper/pencil or word processed) solutions to questions or design documents to indicate your intentions prior to or after coding a programming assignment.

It is expected that you will complete all assignments as an individual unless otherwise instructed (see section on scholastic conduct). If you have questions concerning an assignment, feel free to consult your instructor. In fact, if you spend more than 15 minutes staring at the computer trying to figure out what to do next, *ask for help!* I don't know what you are thinking if you don't tell me. Therefore, take advantage of the fact that I am willing to help you. However, plan your time wisely. Do not wait until the last minute or help may not be available.

Please note: there may be one or more assignments due during the last week of classes.

Exams

There are a total of three exams this semester.

- Two will be "mid-term" exams.
- One will be the final exam.

Approximate dates for the two “mid-term” exams are listed on the class website. Changes to these dates will be posted on the website and clearly announced in class. You are expected to be present for these exams unless you have made prior arrangements. Make-up exams will be offered under very limited circumstances. If you are aware of conflicts prior to the exam, please bring these to my attention as early as possible. These exams are all closed-book exams unless announced prior to the exam. Specific instructions regarding what will and will not be allowed for each exam will be published to the course website prior to each exam, and will be clearly indicated in the instructions for each exam. It is your responsibility to be aware of what is and is not allowed on a given exam.
While I expect that we will assign final grades based on the standard 90/80/70/60 cutoffs, there are a few exceptions:

- I reserve the right to slightly adjust these levels if I deem that certain assigned activities were more difficult than I had expected.
- I reserve the right to lower the final grade of any student who receives less than 50% of the points on the final exam.
- I reserve the right to lower your final grade in the course by one grade level (for example, lower a B to a B-) for each programming assignment for which you fail to submit at least some evidence of effort towards completion. If an assignment has you completely stumped but you gave it some effort, then show me that you made that effort. I can tolerate confusion; I can't tolerate people too lazy to even attempt the assignment.

Class Attendance and Participation

Socrates reportedly said that the best learning takes place on a log, with the teacher on one end and the student on the other. In other words, learning is most effective if it is an interactive process. To be interactive, we must have two things - attendance and participation. Although I will often use the term "lecture," this course will rely less on the "lecture" you may be used to, and more on an open discussion between instructors and students. Regular class sessions will be aided significantly by your participation in discussions, as well as your willingness to ask questions and share your experiences with your classmates.

While I do not take attendance or incorporate it in to your grade I have to strongly emphasize that in a class with limited textbook readings, attendance will play a huge role in your understanding and, thus, your final grade.

Final Thoughts on Grading

I try to accommodate student needs whenever possible, but I can only do so if I know about them. If you ever have to make alternate arrangements for a class session, an assignment, or an exam, contact me in advance. The safest way to make such arrangements is by notifying me via e-mail or phone of your circumstances and of how you can be reached.

All assignments are due at their assigned date and time. In order to receive partial credit, always submit your best effort at that time. Late work may not be accepted for a grade.

Incompletes

Incompletes are awarded only in very rare instances when an unforeseeable event causes a student who has completed all the coursework to date to be unable to complete a small portion of the work in the last week or two of the semester (typically the final project or exam). Incompletes will not be awarded for foreseeable events including a heavy course load or a poorer-than-expected performance. Verifiable documentation must be provided for the incomplete to be granted, and arrangements for the incomplete should be made as soon as such an unforeseeable event is apparent.

Scholastic Conduct

You are responsible for being familiar with the University’s Academic Ethics Policies (https://www.uni.edu/policies/301).

It would be easy if I could just state "copying and collaboration is wrong." But, in fact, that statement is partially inaccurate in a programming environment. There will be many times this semester when I will explicitly ask you to copy some starter code, and there are times when you may (and even should) discuss code with other students. In other words, under certain circumstances, copy and collaboration is not only ok, it is expected.
General guidelines regarding "copying"

- If I provide you with starter code, or if a reading source provides code appropriate to a problem, than "copying" is more than acceptable. In these cases, you should acknowledge the original author of the code in the header block for the code (how to do this will be discussed several times during the first couple weeks of class).
- Copying from other students is expressly forbidden. Doing so on exams and programming assignments will be penalized every time it is discovered. The penalty can vary from zero credit for the copied items up to a failing grade for the course.
- From time to time you may discover an outside source that provides some information/code which helps you tackle a given problem. Any substantive contribution to your solution by another person or taken from a publication should be properly acknowledged in writing. Failure to do so is plagiarism and will necessitate disciplinary action.

General guidelines regarding collaboration

- First and foremost, your final submission for any assignment should be your own individual, original work unless otherwise specified. To do otherwise is to cheat yourself out of understanding, as well as to be intolerably dishonorable.
- That does not mean, however, that you can't talk about assignments with your classmates. If an assignment makes you realize you don't understand the material, ask a fellow student a question designed to improve your understanding, not one designed to get the assignment done. Don't ask to see their answer/code, but ask them to explain how to approach a problem.
- “Working together” on programming assignments is almost ALWAYS considered, largely because it turns in to one student coding and one student copying (see above).

In addition to the activities we can all agree are cheating (plagiarism, bringing unauthorized notes to a closed book exam, etc), assisting or collaborating on cheating is cheating (providing answers to exam questions, discussing test questions with a student who hasn't taken the exam yet). Cheating can result in failing the course and/or more severe disciplinary actions.

Remember: Discussing assignments is good. Copying code or answers is not.

Class Distractions

You may think that you can text/surf the web during class and still follow the lecture, but the research all demonstrates this just isn’t true. Unless we have discussed it in advance (such as in the case of a family medical emergency), all electronic devices should be turned off and left out of sight during class time.

Accessibility

The Americans with Disabilities Act of 1990 (ADA) provides protection from discrimination for qualified individuals with disabilities. Students with a disability, who require assistance, will need to contact the Office of Disability Services (ODS) for coordination of academic accommodations. The ODS is located at 213 Student Services Center. Their phone number is 319/273-2676. Additionally, please contact me immediately if you have a learning or physical disability requiring accommodation.
Guidelines for Success in this Course

Once you have allocated the necessary time for this course, the following suggestions, compiled from student experiences, should help you plan your time use and prepare for the labs and exams:

- Prepare for lecture! Read any assigned readings before the start of lecture.
- Be on time. I will start promptly and will often start with important announcements.
- Make use of the office hours early! Don't wait until late in the term to seek help.
- Be specific in your questions when possible. Rather than asking for an explanation of a large or general topic, be prepared with specific questions or specific examples that raised your questions.
- Write code on your own! Think of simple problems on your own (or from the text) and solve them.
- Realize that this is a class in design as much as it is a course in programming. This is a change in ideas from your first two programming courses. It is no longer sufficient that your code simply runs. Thus, a significant part of your grade will be based on the quality of your code.

Finally, Students are encouraged to use The Learning Center @ Rod Library (formerly The Academic Learning Center) for assistance with writing, math, science, reading and learning strategies. Meet with trained and certified tutors during walk-in hours or by appointment. For more information, visit us in person on the main floor of Rod Library, on the web at tlc.uni.edu or by calling 319-273-6023.