Instructor: Dr. Steven Shaffer. Contact me through ANGEL by emailing ALL COURSE FACULTY. There is an entire team available to support you and you will do yourself a disservice if you email me only.

Important
I have a lot of students in this course; there is no way that I can maintain order and fairness while giving any student special consideration such as handing in material late, sliding you some extra points, etc. If you have a specific disability which needs to be accommodated, you must follow the rules from the office of disability services. Accommodation for late submissions of work are already built into the course structure.

Required Text: Introduction to Programming Using PLEASE by Steven C. Shaffer. Available online at:

http://www.procopyonline.com/

You must purchase a new copy, because it comes with a registration code that will allow you to use the software. Once you’ve paid for the text you can download it more or less immediately.

How your grade is determined:
- PLEASE mini-tests: 3/week, 10 weeks, 10 pts each: 300
- Exercises: 2 points each: 60
- Quizzes: 1 per week, 12 weeks, 4 points each: 48
- Project (assigned week 10): 92
- Total possible points: 500

Letter grades:
- 94% and above: A
- 90%-93.9999%: A-
- 88%-89.9999%: B+
- 84%-87.9999%: B
- 80%-83.9999%: B-
- 78%-79.9999%: C+
- 70%-77.9999%: C
- 60%-69.9999%: D
- 0%-59.9999%: F
**Exercises:** Follow along with the chapters and the videos in the book and type in and run all of the exercises. Upload a screen capture to ANGEL of the program running, similar to those shown in the text. This is the best way to understand the material, and you get some credit for it as well.

**Quizzes:** Quizzes are taken through ANGEL. They are matching problems based on the concepts in the “Test your understanding” section in each chapter. These quizzes are open book but you should not rely on being able to look up the answers during the quiz; if you are not intimately familiar with the material you will not do well on the quizzes. The quizzes have a minute time limit which is enough if you are prepared but not enough if you plan to look up the answers during the quiz. Always verify that your quiz has been received by ANGEL before assuming it is complete.

**Programming mini-tests:** Programming mini-test problems are automatically delivered to you through the PLEASE software. Each problem is unique to you, and you only have a limited time to complete it. There are three of these each week. The text explains how these work and there is also an explanatory video on the ANGEL course site. These problems are graded by testing to see if you can produce the desired output; if so, and you have not “hard coded” the answer (see below), then you will get full credit for it.

**A note on mastery learning:** It's important to remember that if you do not complete a mini-test question in time that YOU CAN ALWAYS GET ANOTHER ONE! If you do not complete a problem within the allowed time frame, simply ask for a new problem. You can keep trying until you get them correct. This is called mastery learning and is very effective. Most of the people who complete this course end up with a 95% or above. The notion of "partial credit" actually works against you, lowering your grade. The mastery learning approach allows you to keep trying until you get it 100% correct. Just keep working through the problems; however, if you run out of time, the best approach is to complete the program anyway - once you complete a few, they become easier. This way you are likely to complete the next problems faster. Sometimes students will be stuck on a certain concept, but keep picking new problems instead of working through one problem to its completion, which is not a good approach.

So, to recap - get a problem and complete it, including asking for help. If it's within the time frame, then submit it. If not, get another one and repeat the process. As you gain experience, the problems become easier to complete.
**Project:** The project is a non-trivial programming project that you will undertake at the end of the semester, which I will assign to you. This project is meant to demonstrate that you have a good grasp of the course material and so it is graded something like a gymnastics event - the harder the program, the better the grade can be. If you do something not very complicated then you will not get better than a “C”. This project is unique to you, and you may not work with anyone else on it. Be sure to see the academic integrity policy below.

**Lateness:** If any submission (exercise, quiz, or mini-test) is late, then you lose 9% + 1% per day it is late. A "day late" starts when the due date/time has passed, even by one second. This is calculated automatically at the end of the semester. Projects are due on the last day of the semester and cannot be late.

**What is a "hard coding"?** When your program runs, it is possible to just have it output the exact characters that are being asked for in order to "match" the expected output. Except for in chapter 1, this does not count as a valid program. Your program must generate the output based on the requirements given, not just "spit out" the expected output. This will become clearer as you get experience with the course.

---

**Academic Integrity Policy**
**Based on the draft policy of the CSE curriculum committee**
**Spring, 2011**

The Pennsylvania State University has high regard for the integrity of the academic process. Consequently, course work that you submit for grading must be your own work, unless the instructor explicitly states that the assignment is to be a collaborative effort. Any collaboration not explicitly permitted by the instructor will be considered cheating. Examples that constitute cheating include but are not limited to:

1. Copying or modifying someone else's work, including copying code from the Internet or textbook (other than the course text or instructor's notes).
2. Receiving help from anyone besides the instructor, the TA, or approved tutors for the course, including help debugging your code.
3. Allowing another student to use your work, giving copies of your code to others, or posting it in a public forum.
4. Writing code for another student.
5. Fabricating program output.
Instances of cheating will result in academic sanctions, which may include an F for the course. In addition, an official Academic Integrity form must be filed with the Judicial affairs office and remains as part of your academic record University-wide. Such reports have implications beyond one course: if your record indicates that you've had prior academic integrity offenses in other courses, you may be expelled from the University.