

# Syllabus

## CS203A: Introduction to Computer Science

### Fall 2015

(Note: this is revised from the paper copy handed out in class. Changes are in red.)

#### Course Information

<b>Course website:</b>	<a href="http://teaching.up.edu/cs203">http://teaching.up.edu/cs203</a>
<b>Instructor:</b>	Dr. Steven R. Vegdahl Email: <a href="mailto:vegdahl@up.edu">vegdahl@up.edu</a> Web page: <a href="http://faculty.up.edu/vegdahl">http://faculty.up.edu/vegdahl</a> office phone: 503-943-7215 office location: Shiley 218
<b>Class Meetings:</b>	MWF 12:30 – 1:35 PM
<b>Classroom:</b>	Shiley 301
<b>Office hours:</b>	M 2:00-3:15pm, T 12:15-1:30pm, W 10:45am-noon, F 8:30-9:45am or by appointment. (If not in office during posted office hours, check in the Shiley 208 computer lab.)
<b>Engineering Fellows:</b>	Nerissa Lemon and/or Caleb Piekstra should be available to help CS 203 students with CS concepts and homework Sunday-Thursday 7-9pm in the Shiley 208 computer lab.

**Bulletin Description:** Create a foundation for computer science and the software development process. Emphasis on good design and programming techniques through practice in writing, running, and debugging programs. Study of a programming language which incorporates objects, structured control statements, classes, inheritance, strong data typing, and sub-programs with parameters. No programming experience expected. (Corequisite: CS 273.)

#### Student Outcomes

At the end of the course, students should be able to:

- Design and construct Java programs that include the following features: input/output, variables, strings, conditionals, loops, and methods
- Design and construct Java programs that use graphics
- Design and construct Java programs that can create, store, and retrieve data from arrays
- Design and construct Java programs that write to and read from files
- Design and construct recursive methods
- Design and implement classes, including the design of instance variables, constructors, methods, and inheritance relationships
- Test programs and fix errors in programs
- Document and comment Java code

These outcomes will be accomplished through:

- Completing homework assignments, reports, in-class activities, and exams
- Participating in class discussion and activities through regular class attendance
- Seeking help of the professor and classmates when necessary

- Communicating ideas orally and in writing

## Course Philosophy

This course introduces concepts related to computer programming and computer science. ***Because this is an introductory course, it is critical that you keep up with the material by completing assignments on time and preparing for each class session by reading the assigned material and completing the Moodle-quiz that covers the readings.*** It's okay to struggle with the concepts, but it is your responsibility to seek help when you are confused, especially since the material continues to build.

**Seeking Help:** I expect you to have questions. You may receive help from classmates (subject to certain constraints--see Collaborative Learning, below) and seek help from the instructor. I encourage you to ask questions during lecture meetings and attend office hours. I am happy to assist you with homework questions if we can discuss your code and questions in person during office hours and other scheduled appointments. I also try to be available frequently via email when I am not on campus. There is free tutoring available by Tau Beta Pi (engineering honors society) members. There

**Start Early:** You will be given one to two weeks to complete programming assignments in this course. For most assignments, you will have seen the topics necessary to complete the assignment the day the homework is assigned. Because the computer is completely literal, it is common to write programs with errors and the computer will not fix the errors for you. The earlier you start on an assignment, the more time you have to seek help, if necessary. Do not start a homework assignment the night before it is due.

**Collaborative Learning:** Your classmates are a huge resource available to you. Because we understand material in different ways, I encourage you to discuss concepts from the course, but ***any code and written work that you turn in must be your own. Unacknowledged copying or using parts of someone else's work, even if it has been modified by you, is plagiarism and is not acceptable.*** When you work with others on homework and projects, *you must acknowledge places where you received help in your homework submissions.* An acceptable way to collaborate is to discuss problems and potential solutions (using paper or a whiteboard) and then writing or coding the solutions at the computer ***on your own.*** When giving help to other classmates, please ***do not give them the answer or show them your own code.*** Instead, ask questions to learn of their understanding and give conceptual explanations—this practice will help you master the material yourself. You may use the instructor, your lab instructor, students enrolled in CS 203, Engineering Fellows, and designated tutors for CS 203 as resources. Remember: you must turn in work that is your own, you must acknowledge the people who helped you, and you are encouraged to seek help when you are confused. Other than starter code for projects, ***any code you submit must be typed entirely by you.***

**Instructor's Responsibility:** In this course, the instructor's job is to guide you in learning about computer programming and computer science. In addition to traditional lecturing, I will have regular discussions and activities during lectures. I expect your full participation and readiness to learn at all class meetings. You should be ready for lecture by doing the assigned readings prior to the lecture. In return, I will do my best to offer suggestions, activities, and explanations to help you learn the material.

**Code of Academic Integrity:** Academic integrity is openness and honesty in all scholarly endeavors. The University of Portland is a scholarly community dedicated to the discovery, investigation, and dissemination of truth, and to the development of the whole person. Membership in this community is a privilege, requiring each person to practice academic integrity at its highest level, while expecting and

promoting the same in others. Breaches of academic integrity will not be tolerated and will be addressed by the community with all due gravity. (Taken from the University of Portland's Code of Academic Integrity.)

## Methods of Assessment

I will assess your learning based on your submitted work, including programming assignments, exams, Moodle quizzes, and in-class activities. Generally, the assignments and in-class activities are your chance to learn, while the exams are the primary way I will assess your understanding of conceptual material.

**Grading Weights:** Course grades will be calculated as follows:

- 45% - Programming Assignments
- 7% - Moodle Quizzes
- 24% - Midterm Exams (3 each at 8%)
- 18% - Final Exam
- 6% - In-Class Activities, Attendance, and Class Participation

**Programming Assignments:** There will be generally be weekly homework assignments, most of which will focus on programming. These assignments are intended to help you learn the material. These assignments should be completed individually, unless otherwise specified. Each programming assignment will be graded on function (whether it works as it is supposed to) and style (whether is readable, well-organized, and concise). I expect you to submit your homework by the due date and time. All programming assignments will be submitted through Moodle (learning.up.edu). (See Late Assignments below for the late assignment policy.)

**Moodle Quizzes:** Moodle quizzes will generally be on the assigned readings for a given week. Each week, they should be available Monday, and are due on Friday before class. You should complete the assigned reading and answer these questions on-line through Moodle.

**Exams:** Exams are intended to serve as learning tools in addition to helping me evaluate your mastery of concepts. There will be three midterm exams and one final exam. The midterms will be given in-class during the regular lecture hours. The final exam is scheduled by the registrar and cannot be moved to a different day or time. ***Please note the exam dates and put them in your personal calendars.*** If you need to miss an exam due to a university-related activity (sports competition, music, ROTC, etc.), you must let your instructor know one week in advance to make arrangements to take the exam early or while you are away from campus. Make-up exams will typically be oral.

**In-Class Activities and Participation:** We will have regular discussions and activities during lecture. Some of these may include items you turn in for credit. Attendance at every class session is expected and attendance counts toward this portion of the grade. If you need to miss a class session, please email your instructor prior to the class session; otherwise, the absence will be considered unexcused, and will affect your grade.

**Late Assignments:** You are granted **five free late days** (5 24-hour periods) to use at your discretion for submitting late assignments without penalty. If you turn in an assignment late, your instructor will presume that you intend to use *late days* unless you advise otherwise via email. Once you have used up your *late days*, your assignment will be penalized 15 points (out of 100) *per day of lateness*. Assignments

handed in more than *three* days late will not be accepted for credit. (You may therefore use at most three *late days* per assignment.)

**Final Grade Assignment:** Course grades will be assigned based on the total points you earn during the semester according to the weighting scheme above. The minimum cutoffs for grades will not change. I do reserve the right to raise your grade, but the following minimum percentages are guaranteed. (For example, if you earn 90% of the points, you will get an A-. If you earn 89% of the points, you earn a B+ but I reserve the right to raise your grade to an A-.)

- $\geq 92.5\%$  A
- $\geq 90\%$  A-
- $\geq 87.5\%$  B+
- $\geq 82.5\%$  B
- $\geq 80\%$  B-
- $\geq 77.5\%$  C+
- $\geq 72.5\%$  C
- $\geq 70\%$  C-
- $\geq 67.5\%$  D+
- $\geq 62.5\%$  D
- $\geq 60\%$  D-
- $< 60\%$  F

**Programming Assignment Grades:** There is no such thing as a perfect program. One way I try to emphasize this is by having a grading scheme where “100” is not special. Programming assignments will generally consist of two parts:

- Basic, “bare-bones” functionality, which all students are expected to do.
- Several ideas for extended functionality.

If you do only the bare-bones assignment, but do it correctly, you can expect a grade of 85 (mid-range B). If you want to receive a higher grade, you should plan on doing one or more of the enhancements.

In grading, every student starts out with a score of 85. Your score is reduced for deficiencies (bugs, poor style/organization, etc.) Your score is increased for enhancements implemented; it may also be increased if your style/organization is particularly good.

One advantage of this scheme is that you will undoubtedly have one or more very busy weeks during the semester, where you perhaps have time to only do the bare-bones assignment, and therefore score an 85. You can make up for an 85 in a subsequent week by doing several extensions in order to receive a score that is above 100.

If you receive a grade above 100% in your CS 273 lab, your score on each homework assignment will be increased by the “CS 273 course grade minus 100”. For example, if your grade in CS 273 were computed at 101.5%, your score on each of your CS 203 programming assignments would be increased by 1.5 points (out of 100).

## Logistics

**Prerequisite:** None

**Corequisite:** CS 273

**Textbook:** The required textbook for the course is *An Introduction to Problem Solving and Programming* by Walter Savitch (seventh edition). [ISBN: 978-0-13-376626-4] The text is available for purchase at the campus bookstore. I ask that you read certain chapters or sections *before* attending the accompanying class session (see the online course calendar for the latest updates to the readings). Lectures are intended to supplement the textbook. I will often put lecture materials online.

**Course Schedule:** The latest version of the schedule is on the course website. The course schedule lists the lecture topics, assigned readings, exams, and due dates for homework assignments and programming projects. It is always subject to change as the semester progresses.

**Lecture Notes and In-Class Programming Examples:** I will make an attempt to put my lecture notes online (course schedule page) before each lecture. Feel free to download (or even print) them if you feel it will help you in taking notes. I will also attempt to put programming examples from class online after each lecture. Feel free to email me a reminder if I forget.

**Disabilities Statement:** If you have a disability and require an accommodation to fully participate in this class, contact Accessible Education Services (AES), located in Buckley Center 163, as soon as possible. If you have an AES Accommodation Plan, you should make an appointment to meet with me to discuss your accommodations. Also, you should meet with me if you wish to discuss emergency medical information or special arrangements in case the building must be evacuated.

**Assessment Disclosure Statement:** Student work products for this course may be used by the University for educational quality assurance purposes.

**Green Dot Statement:** University of Portland Faculty, Staff, and Students are committed to creating a community free of interpersonal violence, in which all members feel safe and respected. Each of us has a personal responsibility to reject violence or intimidation of any kind. Resources for those experiencing or wishing to report violence can be found on our *Community Against Violence* website: <http://www.up.edu/cav/>.

**Improving the Course:** I welcome your feedback about the course at any time. I may ask for your feedback periodically and you will have the opportunity to evaluate the course at the end of the semester.