CS 201.01: Computer Science I
Computer Science Department
Loyola University Maryland
Spring 2015

Instructor: Dr. Megan M. Olsen  Office: Donnelly Science 124
Office Phone: 410-617-2852  E-mail: mmolsen@loyola.edu
Office Hours: TBD! Take the poll on Moodle, 2nd week I’ll announce final office hours
And other times by appointment.

Class: MWF 10:00 - 10:50 (Knott Hall 006)
Lab: Tues 10:50 – 12:05 (Donnelly 121)

Moodle Course Website: http://moodle.loyola.edu
CS Department Website: http://www.cs.loyola.edu
Instructor Website: http://www.cs.loyola.edu/~olsen

Catalog Description: A general survey of the major areas of computer science including theory of computation, elementary digital logic, programming languages, artificial intelligence, common application software, ethical issues in computing, and software design. Introduces elementary structured programming, including top-down design, object-oriented design, functions, loops, and arrays. First course in the major’s sequence.

Prerequisites: None

Pearson’s MyProgrammingLab website (http://www.myProgrammingLab.com)

Other Resources:
- Java documentation (from Sun): http://download.oracle.com/javase/7/docs/api/
- jGRASP download site and documentation: http://www.eng.auburn.edu/grasp/
- Java tutorial: http://chortle.ccsu.edu/CS151/cS151Java.html (slow, methodical, and very comprehensive)
- GitHub: http://www.github.com

COURSE OBJECTIVES:
Upon successful completion of this course, the student should be able to:
1. Understand the basics in the breadth of computer science, such as ethical issues in computing, the hardware of computers, and human computer interaction
2. Design solutions to problems that can be solved using programming, via algorithms, flowcharts, and object-oriented design
3. Understand Java syntax well enough to use Java programming to solve problems involving:
   - Mathematical computation
   - Decision making
   - Repetition
   - Java classes
   - Arrays
4. Use an Integrated Development Environment (IDE) to enter, compile, and run simple programs
5. Use and create program documentation
6. Understand the meaning of programs and test their correctness appropriately
COURSE OUTLINE: The schedule is subject to change as the need arises.
Most classes will focus on Java topics. After finishing a Java topic, we’ll devote a day to a “breadth-first” CS topic, i.e. something not specific to Java programming. Breadth-First Topics are denoted in italics in this syllabus. Tuesday labs are normally devoted to an exercise that will help you learn programming. The lab period is also used for tests.

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Major Topics</th>
<th>PA/Quiz/Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/12-16</td>
<td>Introductions to each other Introduction to the course</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B: Introduction to computing Variables &amp; Coding Style</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1/19-23</td>
<td>University Closed Monday – No class! Arithmetic expressions and operations</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>User input GitHub and code version control</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1/26-30</td>
<td>B: Human Computer Interaction Making decisions: selection</td>
<td>PA1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Discuss PA2</td>
</tr>
<tr>
<td>4</td>
<td>2/2-6</td>
<td>More selection techniques Flowcharting, control flow, and testing</td>
<td>Quiz 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B: How data is stored</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2/9-13</td>
<td>Java applets and graphics</td>
<td>PA2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B: Data correction, encryption, and compression</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2/16-20</td>
<td>Repetition</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B: Cloud Computing</td>
<td>Discuss PA3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz 2</td>
</tr>
<tr>
<td>7</td>
<td>2/23-27</td>
<td>Repetition</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3/9-13</td>
<td>Methods</td>
<td>PA3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TEST 1</td>
</tr>
<tr>
<td>9</td>
<td>3/16-20</td>
<td>B: Artificial Intelligence Classes</td>
<td>Quiz 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Discuss PA4</td>
</tr>
<tr>
<td>10</td>
<td>3/23-27</td>
<td>Classes</td>
<td>PA4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B: Computer Hardware</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>**Deadline to withdraw is Friday</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>**Science Grant Seminar is Thursday</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3/30-4/3</td>
<td>Arrays</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>University Closed Thursday &amp; Friday – No Class!</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4/6-10</td>
<td>University Closed Monday – No Class! Working with Arrays</td>
<td>Quiz 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PA5</td>
</tr>
<tr>
<td>13</td>
<td>4/13-17</td>
<td>Searching &amp; Sorting in Arrays</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Discuss PA6</td>
</tr>
<tr>
<td>14</td>
<td>4/20-24</td>
<td>More on Sorting in Arrays</td>
<td>Quiz 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B: Ethics in Computing</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>4/27</td>
<td>Exam Review</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>** We will have an extra optional exam review Tuesday afternoon in addition to the in class review</td>
<td>PA6</td>
</tr>
</tbody>
</table>

Final Exam on Wednesday 4/29, 9AM, KH006
GRADING:
Your final grade will be determined by the number of points that you earn and will be calculated as below and saved in Moodle:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collected assignments</td>
<td>20%</td>
</tr>
<tr>
<td>Labs</td>
<td>15%</td>
</tr>
<tr>
<td>4 quizzes</td>
<td>10%</td>
</tr>
<tr>
<td>6 programming assignments</td>
<td>30%</td>
</tr>
<tr>
<td>1 test</td>
<td>10%</td>
</tr>
<tr>
<td>1 final exam</td>
<td>15%</td>
</tr>
</tbody>
</table>

Final letter grades will be calculated on a scale close to the following:

- A = 91 - 100%
- B = 81 - 84%
- C = 71 - 74%
- D = 60 - 64%
- A- = 88 - 90%
- B- = 78 - 80%
- C- = 68 - 70%
- F = 0 - 59%
- B+ = 85 - 87%
- C+ = 75 - 77%
- D+ = 65 - 67%

1. Tests and quizzes are announced. Consult Moodle for final schedule. Tests occur during lab time.
2. Homework is due at the beginning of class. Generally, half the points are awarded for completeness and half for correctness. Correctness is determined by my checking all or a portion of the assigned exercises.
3. Assignments to be submitted on paper should be submitted on standard size paper. Please make them legible and neat. *Illegible and/or sloppy assignments will NOT be graded.*
4. Any assignment, PA, or lab with multiple sheets of paper MUST be stapled or paper clipped together.
5. Programming assignments (PAs) will be given periodically. At least a week is given for a PA. PAs are due at the beginning of the regular class period. See Moodle for dates.
6. If you believe you lost points in error on any assignment, you must bring a written explanation to my office of why you disagree with the grading *within 2 weeks of the assignment being returned/discussed in class.* However, feel free to come by for clarification about any assignment at any time.

Labs:
- Labs will be conducted using pair-programming, and should follow the lab guide.
- Bring your book and notes to lab each week for reference.
- Labs will be due in class the following Monday. You may turn it in early if it’s complete.

Late Work: You have 2 late passes you may use for any assignment or programming assignment. Each pass allows you to submit the assignment at the beginning of the next class period without any penalty. Just label the assignment (at the top of the first page) with "Use a Late Pass." If it is a GitHub assignment, create an issue titled “Use a Late Pass.” You may use both late passes for 1 assignment, allowing you to submit 2 classes late.

If you turn in work that is late without a late pass, for every 8 hours late I will deduct 5% from your final earned grade on that assignment (so, 11 hours late will be -10%); this includes hours over the weekend. This is the equivalent of 15% a day, but allows you to save some off that if you really only need a few more hours, or just forgot to bring it to class.

If you wish to turn it in after hours or when I am not on campus, you must submit it to me electronically in a legible format. Typing assignments will make this easy to accomplish. You may ONLY submit an assignment via e-mail if it was meant to be turned in on paper during class; otherwise, submit it via the stated electronic method (Moodle or GitHub).

Of course, sometimes things happen. If you have an excused reason for needing more time (sickness, death of a relative, etc) come talk to me BEFORE the day the assignment is due and we can discuss your options. **Unless you have an excused absence, homework may not be submitted more than 1 class meeting late, as we will discuss the answers in class after that date.**
Course Technology:

- **Moodle**: Grades, assignment deadlines, and handouts will all be in Moodle. Also, all preparations for class will be assigned and submitted here. All course meetings will be recorded through Panopto, which can be accessed through our Moodle page.

- **MyProgrammingLab**: This tool allows you to practice programming with instant feedback. You may keep trying until you get it right. Specific problems will be assigned for homework.

- **GitHub**: All programming assignments after week 1 will be submitted via GitHub. GitHub will store your files online (but privately – only you and I can see them), so you can use it to back up your files. You can also use GitHub to ask me about a problem you are having with your programming assignment or lab code by creating an “issue” in the repository.

Inclement weather policy:

If Loyola is opening at 11AM or later (including being closed for the day) on a lecture day, we will have a virtual class. Class will be conducted via Panopto, which you can watch in real time through our Moodle class page. You will be expected to take notes, and will be held responsible for all material covered as if it was an in-person class meeting. I will post the lecture notes on Moodle in time for you to print them before class, which I highly recommend. If you do not have internet access during class time, you must watch the lecture at a later date but before the next class meeting.

If we miss a lab due to a snow day you will be expected to either watch a lecture instead, or perform the lab on your own time (this can be with a partner). I will inform you via email which option we will be taking.

EXPECTATIONS:

**What you can expect of me**: You can expect that I will come to class prepared and excited to help you learn computer science. You can expect me to be knowledgeable about the material. You can expect me to be available during my office hours and at other times that you arrange to see me. I am often on email if you can’t make it to my office or if it is the evening! Expect me to return graded work promptly. You can expect me to treat you respectfully. Please see me privately if you think that is not the case.

**What I expect of you**: Computer science is a field requiring creation and activity, and works best when you do more than sit back and listen. *You need to DO it!* I therefore expect you to come to every class and lab prepared to contribute—this means that you need to complete the assignment, do the reading, and determine what you need help understanding. You contribute to class by intelligently questioning the instructor and offering further explanation to your classmates and me. I expect you to take responsibility for learning computer science—you won’t be sorry. Further, I expect you to treat every other member of the class (including me) with respect.
PROCEDURES AND TIPS FOR SUCCESS:

0. This is a 4-credit course. You should expect this course to claim more time and effort than a 3-credit course (you can expect to invest at least 4 x 3 = **12 hours per week outside of class time**).

1. If you have a disability of any type that requires an accommodation, please let me know as soon as possible. The disability forms are now being emailed directly to the professor; if you’ve had a form emailed to me, please set up a time for us to discuss your accommodations. To request academic accommodations due to a disability, please contact Disability Support Services (DSS), Newman Towers West 107, at DSS@loyola.edu or call (410) 617-2750 or (410) 617-2062.

2. **Be Prepared:**
   a. You are expected to have read or watched the preparation material listed on Moodle *before* the lecture. The textbook presents material in a very concise manner. You will greatly enhance your class experience by coming prepared.
   b. Many classes will have small exercises for you to perform before class based on the preparation. This is designed to help you stay on top of the course material, and will affect your assignment grade.

3. **Regular on-time attendance is necessary for success in this course.** As a point of courtesy, plan to arrive on time for class. Students are responsible for material presented and assignments made during absences. Normally, make-up exams are NOT administered. **Attending lab is NOT optional.**

4. **Laptops are not allowed in class.** We have computers at each desk for your use.

5. **This is a "hands-on" course.** Therefore, besides reading the texts, attending all classes, taking good notes, completing assignments, and studying, to be successful you will need to spend a substantial amount of time in the lab or at your computer. Many students find they need five to eight such hours per week.

6. **Start assignments EARLY.** Try to avoid pulling “all-nighters” to complete programming assignments. No one thinks clearly or uses time effectively when they are exhausted. Start planning a programming assignment when it is first distributed. Remember that the Intro Programming Lab (DS121) is usually quiet and not crowded.

7. **Should you need extra help, please see me during office hours or make an appointment for a mutually convenient time.** Do NOT wait until you are totally lost and have failed a test or quiz. Your success is my first priority. Even if I am busy, please know that I am determined to be available to assist you.

8. **Collaboration:**
   All assignments that are not pair programming exercises are to be **completed by the student as an individual.** Collaboration on algorithm design is encouraged, and **if you collaborate on design you MUST note that at the top of the assignment. However, sharing code or answers to exercises is NOT allowed.**

Academic honesty is required of all Loyola students at all times. Students are expected to abide by the Honor Code:

The Honor Code states that all students of the Loyola Community have been equally entrusted by their peers to conduct themselves honestly on all academic assignments.

The students of this College understand that having collective and individual responsibility for the ethical welfare of their peers exemplifies a commitment to the community. Students who submit materials that are the products of their own minds demonstrate respect for themselves and the community in which they study.

All outside resources or information should be clearly acknowledged. If there is any doubt or
question regarding the use and documentation of outside sources for academic assignments, your instructor should be consulted. Any violations of the Honor Code will be handled by the Honor Council.

Any copying of an assignment, whether electronically or by hand is considered plagiarism. Students submitting non-trivial projects with identical structure will be considered to have acted dishonestly. Such students may be referred to the Honor Council for disciplinary action. At the very least, two or more students presenting assignments identical in all important aspects will share the points from a single grade.

As soon as code is exchanged, the line between collaboration and plagiarism has been crossed. Be aware that I use an automated tool that determines the degree of similarity of programs very effectively.

Examples of what I would consider OK:
1. Discussion among students on how to approach a program, such as understanding the problem and requirements of the assignment
2. Discussion among students on algorithmic design or testing plans, but only if no written notes leave the discussion (i.e. you only leave that discussion with your own ideas in your mind)
3. Discussion among students on how to debug an error in a program
4. Working together with your partner on a pair-programming assignment such as lab

Examples of what I would consider cheating:
1. Exchanging code either via hardcopy or electronically
2. Taking another student’s code with or without their knowledge
3. Dictating to another student how to write their code (This is analogous to dictating sentences that someone should write in an English paper).
4. Using code from any other source (WWW, a friend, etc)
5. Showing another student your work as a way to help them figure out their own assignment

You are always welcome to ask if a particular type of collaboration is OK. Please do so before engaging in a collaboration you are unsure about. Plagiarism inhibits the learning of its participants.

You will be asked to sign the following pledge that will appear at the bottom of all tests and quizzes:

“I understand and will uphold the ideals of academic honesty as stated in the Honor Code.”