Class Meetings @ BP 326 or BP 310

<table>
<thead>
<tr>
<th>Section 02:</th>
<th>Mohammad Hadian, Instructor <a href="mailto:mhadian@clarku.edu">mhadian@clarku.edu</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon, Wed 12:00 pm – 1:15pm</td>
<td>Please always include “CS121” in the subject.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lab Meetings @ BP 310:</th>
<th>Teaching Assistants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab 902 Wed 6:00pm – 7:15pm</td>
<td>Mariah Papy (<a href="mailto:MPapy@clarku.edu">MPapy@clarku.edu</a>)</td>
</tr>
<tr>
<td></td>
<td>Akhmad Kurbanov (<a href="mailto:AKurbanov@clarku.edu">AKurbanov@clarku.edu</a>)</td>
</tr>
<tr>
<td></td>
<td>Daria Manea (<a href="mailto:DManea@clarku.edu">DManea@clarku.edu</a>)</td>
</tr>
<tr>
<td></td>
<td>Tutoring hours held in the CS lab, BP 310.</td>
</tr>
<tr>
<td></td>
<td>The CS lab will generally be open Sunday – Thursday nights.</td>
</tr>
</tbody>
</table>

Course Description
The Data Structures course is a 2nd semester continuation of CS 120, Introduction to Computing. The course features an emphasis on such topics as abstract data types, collections, and dynamic data structures such as linked lists, stacks, queues and binary trees. The usage and implementation of recursion is also discussed, as is the elementary analysis of algorithms. This course is designed to continue your development of computer programming skills, introduce you to common building blocks of computer programs, and help you develop problem-solving skills in computing by applying those building blocks.

It should be noted that this is not a course about Java. The big ideas in this course include data abstraction, algorithms, and efficiency. Java is the computer language we use to investigate these concepts.

Teaching/Learning Method
This course will follow a rigorous schedule of assignments. Each assignment corresponds to a section of the course content and textbook. Failure to keep up with the assignments will likely result in you failing the course. Each session builds on the prior session and is a required building block for the following session. It is very difficult for you to be successful in the course if you miss class sessions or assignments.
Textbook
1) Computer Science: An Interdisciplinary Approach 1st Edition
by Robert Sedgewick, Kevin Wayne

2) (optional) Introduction to Programming in Java: An Interdisciplinary Approach (2nd Edition)
by Robert Sedgewick and Kevin Wayne

Other online readings and tutorials will be posted to the schedule page.

Programming Language and Environment
Java is a programming language for this course.

There are several Java IDEs, including IntelliJ, BlueJ, Eclipse, and Dr. Java. Mainly we will use IntelliJ for this course. You are free to use any IDE you are most comfortable with. IntelliJ is installed on the CS310 lab computers, You may also use your own computer for CS121 course work.

Grading
The following percentages are tentative and may be changed at our discretion at any time:

- Weekly Exercises and Class: 20%
- Programming Assignments (about 6): 30%
- Lab Programming Exams (about 2): 20%
- Written Final Exam: 25%
- Attendance, Class Participation, Extra Credits: 5%

Withdrawing from the Course
If you feel that you want to drop or withdraw from the class, please come talk to your instructor about it as early as possible; we want to help you succeed, but you need to ask for help.

The last date to withdraw and receive a "W" grade is in late March.

What You Need to Know About Computer Programming
We believe anyone can succeed at learning to program. This is a second course in computer programming, and as such it is expected that students will have familiarity with writing basic computer programs in the Java programming language.

In addition, you will need time, and this is more important than you can imagine. Following Clark curriculum guideline, each student is expected to spend about 180 hours on this course. This means about 8 hours out of class study time every week.

Many people believe that computer programming is extremely difficult, and that the code is written in some arcane syntax understandable only by experts. Although some parts of the process are indeed complex, most of the source code required for weekly exercises and homework assignments can be easily understood.

So, what makes programming so hard? It's not the difficulty: It's the time required to achieve any decent results. The homework assignments will take time; so make sure you have plenty of it.

Adapted from text in “Core Techniques and Algorithms in Game Programming,” Daniel Sanchez-Crespo Dalmau, and Aaron Stevens (BU)
Policies and Miscellaneous

The official administrative business of this class will be conducted by email.
Grade questions/disputes, explanation of absence, etc. will be processed via email so that we both have a written record of what was agreed. Feel free to discuss in person but an email follow-up is required for the official record.

Attendance and discussion/asking questions are expected and will be reflected in your grade.
If you must be absent, please email me in advance to let me know why you won't be in class, and to let me know what you will do to keep up with the assignments. CS 121 is not a correspondence course.

Lab attendance and submission of the lab work is required.
Please attend your scheduled lab section. Lab work is an essential part of your learning. There is no more valuable time available to you than time in lab, so don't waste it doing other things – in particular spending your time looking at your phone is very strongly discouraged.

Assignments are due on the time and date stated on the assignment page on the web.
- Each student is allowed to submit up to two assignments up to 48 hours late, without penalty.
- No other late submission will be accepted, except for extraordinary situations.

Plan your work accordingly, and work on all assignments as soon as they are given so you can ask questions in class and get assistance in the labs and tutoring hours.

Students are responsible for ensuring that assignments are correctly submitted. If you have a question or problem, seek help from CS121 staff immediately.

No special make-up work will be accepted after the end of the semester. Don't even ask.
In the event of a documented major medical problem, a grade of Incomplete will be given pending the submission of complete work. However, make up work “to improve one’s grade” will not be accepted.

It is the student’s responsibility to retain all papers, quizzes, and exams that have been graded and returned. Should these original documents not be available in the event of a grade dispute, we will need to defer to our own records.

Grades are not negotiable. Don’t even ask – just do the work and you’ll get the grade you deserve. Of course, please bring any clerical grading errors to my attention by email and I will gladly fix them.

Academic Integrity
It is the student’s responsibility to know and understand the Clark University Academic Integrity policy, which is within the Academic Advising Handbook (The Blue Book) available at the Academic Advising Center and online at http://www.clarku.edu/offices/aac/integrity.cfm.

Pair Programming*: Some cs121 assignments allow pair programming. When partnering, both students work together (in the same room) and discuss, write, debug, test, analyze, document, and submit all elements of the assignment. In this case, only one partner (with the other partner present) submits their assignment code and report; the other partner submits only an abbreviated readme.txt that contains both partners’ names and logins. Both partners are responsible for understanding all parts of the submitted assignment and receive the same grade.

Collaboration policy*: Programming is an individual creative process much like composition. You must reach your own understanding of the problem and discover a path to its solution. During this time, discussions with other people are permitted and encouraged. In fact, we recommend student cooperation in understanding programming concepts, algorithms and system features. You are encouraged to discuss lecture materials, textbook examples, labs, assignment problem specifications and expected outputs, and to seek and receive help with the Java Programming Language, BlueJ IDE, Eclipse IDE, and other tools. However, when the time...

CSCI 121: Data Structures Syllabus: Spring 2019
comes to write code that solves the problem, such discussions (except with course staff members) are no longer appropriate: the code must be your or your pair’s own work. For each assignment, you must specifically describe in your readme or report file, whatever help (if any) that you received from others and tell us the names of any individuals with whom you collaborated. This includes help from friends, classmates, lab TAs, and course staff members.

_Do not, under any circumstances, copy another person’s code._ Incorporating someone else’s code into your program in any form is a violation of academic regulations. In addition to the definition of plagiarism in the handbook, _with respect to CS121, plagiarism is specifically defined to include (but is not limited to) the following:_

- copying any part of someone else’s assignment/program, even if you have permission and/or have modified the code
- sharing or giving your assignment/code or even a subset of your assignment/code to another student
- reviewing another student’s solution (including from past semesters)
- reviewing solutions on the internet

Also you may not publish solutions to cs121 programming problems in a way that could compromise their utility as pedagogical tools.

It is our policy to use automatic plagiarism detection software, and suspicious similarities will be uncovered.

If you have any questions about these matters, please consult a course staff member.

Alleged violations will be reported to Clark College Board for review. If found guilty, a violator will receive an assignment/exam grade of zero for the 1st offense, a course grade of F for the 2nd or later offense, plus whatever disciplinary action the College Board imposes.

*Parts of the text are adapted from course webpages for “COS226 -- Data Structures and Algorithms”, R. Sedgewick and K. Wayne (Princeton)*