CSC/CPE103: Fundamentals of Computer Science III
Fall 2014
Course Syllabus

Instructor: Dr. Hasmik Gharibyan
Office: 14-207
Phone: 756-7068
E-mail: hghariby@calpoly.edu

Course Web-site: http://www.csc.calpoly.edu/~hghariby/CPE103

Office hours: Mondays: 1210 – 1310; Wednesdays: 1210 – 1 and 210 – 3; Fridays: 1210 – 130
Note: office hours may be re-arranged to allow more time on project due-days and
the day before an exam

I. Textbook: “Data Structures and Algorithm Analysis in Java” by Mark Allen Weiss, third

Notes: (1) you are not required to purchase the textbook; (2) second edition of this textbook is acceptable

II. Prerequisites: The following two prerequisite requirements are strictly enforced:

1. Grade C- or higher in CSC/CPE102: Fundamentals of Computer Science II.
   You are expected to be knowledgeable and experienced in problem solving and software
development using an object-oriented programming language (currently Java).

2. Grade C- or higher in CSC141: Discrete Structures I.
   You are expected to know and be able to apply mathematical concepts, formulae, and proof
techniques learned in CSC141.

   For a quick review of the prerequisite material read Chapter 1 of our textbook.

III. Class Objectives:

- To learn to define and implement classic data structures.
- To learn classic algorithms for basic operations on different data structures.
- To implement and test classic algorithms for basic operations on different data structures.
- To learn to analyze and evaluate algorithm performance and compare different algorithms.

IV. Major Topics Covered in the Course:

- Abstract Data Types (ADT), Collections, Data Structures (chapter 3).
- Algorithm performance analysis (chapter 2).
- Stacks and stack operations (chapter 3).
- Queues and queue operations (chapter 3).
- Priority queues; binary heaps and heap operations (chapter 6).
- Binary Search Trees and BST operations (chapter 4).
- AVL Trees (chapter 4) //the extent of coverage depends on the available time
- Tree traversals; implementing iterators for different binary tree traversals (chapter 4).
- Hash Tables. Hash Table operations (chapter 5).
- Graphs and graph algorithms (chapter 9).
- Efficient sorting algorithms and their analysis (chapter 7).
V. Course structure:

**Lectures**: introduction to various data structures and related algorithms, their implementation and analysis. Your presence at lectures will not be checked (except the first day of classes). However, you are strongly advised to attend lectures – experience shows that regular attendance helps students to perform better on exams. Besides, the lab assignment following a lecture may be directly connected to the topic of that lecture.

If you missed a lecture, **it is your responsibility to**:
1. Obtain lecture notes from your classmates and lecture handouts from me
2. Find out what announcements were made in the classroom

**Lab sessions**: during these sessions you will be working with a partner on small programming assignments (for credit) – you will implement data structures and related operations, as well as small applications for these structures. Typically, you will get two lab assignments per week (first one on Monday, the second one on Wednesday) – both assignments need to be completed and demoed by the end of that week’s last lab session (on Friday).

1. **Lab partners**: You will be asked to work on lab assignments in teams of two (in pairs). The main purpose of pair setting is to provide a better learning environment where students have the opportunity to discuss the assignment with someone else, to brainstorm and learn from each other. Research shows that with pair programming students learn better and work more productively. For details on how to work in a pair setting and to learn its advantages, check the video created by NCSU at [http://www.youtube.com/watch?v=rG_U12uqRhE](http://www.youtube.com/watch?v=rG_U12uqRhE).

**Here are some basic rules and guidelines:**

- In pair programming environment the two students share a single workstation and each is assigned a role: “driver” or “navigator”. The navigator is the person who leads the problem solving effort. The driver is the person who operates the keyboard (of course the driver may assist in solving the problem). Members of the team should rotate – each partner should take the “driver” role roughly 50% of the time the pair is working together. The objective is to work together and learn from each other, NOT to divide the work into two pieces with each partner working on a different piece.

- **You may NOT touch the code without your partner.** This means you can’t alter a single character of your program unless you are working as a pair. You may read the problem description individually. You may think on how to attack the problem when you are alone. But you may not write down anything or do any work on the computer unless you are both working together.

- Only ONE demo needs to be made for a pair; both partners’ presence is required at the demo.

- You may team up with different people for different assignments. However, you should not change partners during (after the start of) the week unless the professor authorizes it.

- **If one partner is late or absent (maximum 5 minutes wait)**, the other partner will continue the work on that assignment solo or will be teamed up with someone else. The late/absent partner will be allowed to work on the current assignment alone but only for 75% of the credit for that assignment.

- If your team is dysfunctional, talk to the professor without delay.

- If you are left without a partner, talk to the professor without delay.
- It is important that each team member has a copy of the work at the end of each lab. Do NOT assume that your partner will be available whenever you need the code (e.g. next lab). Also, note that some of the code will be used in your future lab/project assignments.

**ATTENTION:** The goal of teamwork will NOT be achieved and the opportunity will be wasted if one partner does the work and the other watches silently. Thus:

- **DON’T be a passive teammate** – you will not only hurt yourself, since you will not receive the expected learning experience and practice, but you will also hurt your partner by taking away the experience and the value of teamwork.

- **DON’T be a bossy/intimidating teammate.** Allow your partner to contribute and feel valuable to the team. Don’t deprive both of you the experience and value of teamwork.

2. **Getting Credit for Lab assignments:** Typically there are 2 lab assignments per week – first assigned on Monday, second assigned on Wednesday. Each assignment will receive its individual credit. To get credit for an assignment, a team needs to **demo** its work **in person** during a **lab hour**: you will be given a sheet with “demo instructions” where you will be asked to run your program(s) for the specified set of test cases. When demoing your work, you may also be asked to show your code and/or run additional test cases. Note that:

- **One demo** needs to be done for a team; **both partners’** presence is required at the demo. If one partner is not present during the demo, he/she will not get credit for the assignment(s) unless they can manage to demo the assignment(s) on their own before the deadline.

- The **deadline** to demo week’s assignments is the **end of the last lab session of that week**.

- You are encouraged to demo your work **as soon as you finish** and not wait till deadline.

- If an assignment is not demoed **by the deadline**, you won’t get credit for it.

**The credit for a lab assignment is given as follows:** if your program works correctly on all test runs, you will get the full credit. If your program compiles, executes, and produces results that are not all correct, you will get a partial credit (in increments of 5%) appropriate to the degree of correctness of your work. If your program doesn’t compile, or compiles but doesn’t produce any sensible results, you will NOT get credit for it (a grade zero will be recorded) – students entering CPE103 should be able to write code that at least compiles and produces some sensible results.

**Note:** If the demo establishes flaws in your program, make sure to correct them since that code (or some parts of it) may be used in your future assignments.

3. **Attendance:** you are **required** to attend lab sessions. Your attendance will be checked and will be factored in the calculation of your weighted grade average for the course.

**Important things to remember:**

- If you are late, you will get penalized for your attendance (proportional to the lost time);

- If you are more than 5 minutes late or if you miss the lab, in addition to the penalty for attendance, you will also receive **25% penalty on the current assignment.** After 5 minutes wait, your partner will start the work solo or will be paired with another partner.

- You may leave the session early (or skip the Friday’s session) without penalty **ONLY** if you have completed, demoed, and received credit for **ALL assignments** of that week.
**Programming assignments (projects):** during the quarter there will be 5 larger programming assignments. The full credit for each assignment is 100 points. You will be given at least one week to complete each assignment. The deadline for each assignment will be stated on the assignment handout.

You are encouraged to work on project assignments in a team of two (typically with your lab partner) following the rules and guidelines described above in the subsection “Lab partners” (pages 2-3). If you choose to work on a project alone, let the professor know so she can pair your lab partner with someone else.

Here are some rules for these projects:

1. You are required to submit the **electronic** copy of your work (instructions on how to submit your work via *handin* will be given with your first assignment).

2. There should be **only ONE submission** made for a team, with **BOTH names** listed in the headers of **all** files; the submission should be made into only one teammate’s account.

3. All assignments are due by the deadline stated on the assignment sheet.

4. A late submission will be accepted **only** within 24 hours after the deadline; however, a penalty (50 point deduction) will be applied.
   Assignments turned in later than 24 hours from the deadline will NOT be accepted.
   **Note:** In case of emergencies other arrangements may be made – talk to me ASAP.

5. Programs that do not compile or do not produce any sensible results will **NOT get any credit** (a grade **zero** will be recorded) – students entering CPE103 should be able to write code that at least compiles and produces **some** sensible results.

6. Partially complete programs will be accepted, and if they compile, execute and produce sufficient amount of sensible results, they will receive partial credit. Thus, it is better to submit a working incomplete program than a complete but non-working one.

**You can collaborate ONLY with your teammate (if you have one). Any other collaboration is prohibited; this includes helping another student or getting help from another student.** Violation of this rule will be treated as a case of **academic dishonesty** for all students involved (see the Cheating Policy on page 6).

Seek help **ONLY** from your professor or from tutoring center (see page 5)

**Attention:** ALL students involved in cases of cheating, plagiarism, or academic dishonesty will be severely punished. This applies not only to those who cheated or copied, but also to those from whom the work was copied.

**You are responsible for protecting your work from copying.**

**Exams:** There will be three “closed book” exams during the quarter:

*Midterms:* two 1-hour exams during the quarter.

*Final:* a comprehensive 3-hour **common** exam (typically on *Monday evening* of finals’ week).

**Attention:** There will be **NO** makeup for missed exams (a grade **zero** will be recorded). For **documented special cases and extreme emergencies** other arrangements may be made (inform the instructor about the emergency as soon as possible).
VI. Grading: Your weighted average grade for the course will be calculated as follows:

<table>
<thead>
<tr>
<th>Assignment/test</th>
<th>number</th>
<th>total weight</th>
</tr>
</thead>
</table>
| Labs:
  assignments         | 16-18  | 10 %         |
  attendance            | -      | 5 %          |
| Projects              | 5      | 15 %         |
| Midterm exams         | 2      | 35 % (17.5 % each) |
| Final exam            | 1      | 35 %         |

Note: All lab assignments are of equal value and all project assignments are of equal value.

Your final grade will be assigned as follows:

<table>
<thead>
<tr>
<th>Weighted Average</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>≥ 90%</td>
<td>A- or better</td>
</tr>
<tr>
<td>≥ 80%</td>
<td>B- or better</td>
</tr>
<tr>
<td>≥ 70%</td>
<td>C- or better</td>
</tr>
<tr>
<td>≥ 60%</td>
<td>D- or better</td>
</tr>
<tr>
<td>&lt; 60%</td>
<td>F</td>
</tr>
</tbody>
</table>

VII. Tutoring:

The CSC study support center offers free assistance on an individual basis to students in the introductory CSC courses: CSC/CPE 101, 102, 103 and 141. Tutoring is provided by carefully chosen upper-class students from the CSC and CPE majors. The center provides assistance Sunday through Thursday at evenings. Times and locations will be announced around 2nd week of classes – look for flyers in the 14 building’s hallways and announcement boards.

VIII. Office hours:

I encourage you to come to my office for help during office hours (office hours are listed at the beginning of this handout). You can come to the office hour with any problems you may be having in this course, including the lecture, the lab assignments, and the project assignments. You should come prepared with specific questions so we can get right to the point.

- If you need help with any form of programming activity, you should have all relevant files available.
- If you have questions about lecture topics, be sure to bring your class notes.
- If you missed a lecture, **don’t expect me to redeliver the lecture for you** – you must learn the material on your own and come to me with specific questions.

If there are other people waiting, **I may have to limit the amount of time I can devote to each student**. In such cases you are of course welcome to get back in line for another turn.

*The e-mail or the telephone is NOT a substitute for an office hour.*
IX. Electronic Announcements:
Once in a while I send e-mails to the whole class with announcements on the coursework, as well as clarifications/notes on an upcoming test or assignment. These e-mails are addressed to your Cal Poly account so make sure to check your account often or forward your mail from that account to the one you use on daily bases.

X. E-mail:
Use e-mail to report a problem, to inform about a special personal situation, to ask for an appointment. Do NOT email questions seeking help in coursework – do NOT view e-mail as a substitute for an office hour. Do NOT send code with expectation to get help in debugging.
I will provide all the help you need, but it has to be in person.

Attention:
1. Don’t expect your email to be answered the same day.
2. Don’t expect your email to be read on weekends or on late hours of weekdays.

XI. Classroom civility:
The following is considered disruptive behavior and should be avoided:
- Arriving late
- Cell phones going off in class
- Packing up before the class is dismissed
- Eating in class
- Chatting with neighbors
- Sleeping in class

XII. Cheating Policy:
Academic dishonesty, cheating, and plagiarism are considered serious violations of expected student behavior. See the definition of these acts, as well as Cal Poly rules and policies on these matters at: http://www.academicprograms.calpoly.edu/academicpolicies/Cheating.htm

Students involved in such cases (ALL parties) will be reported to the Office of Student Affairs and will be expelled from the course with a grade F.

Note: You are responsible for protecting your work from copying.