CSE 232 Fall 2015

Shane Carr

August 28, 2015

1 Introduction

Welcome to CSE 232! This course helps bridge the gap between algorithm theory and practical problem solving. The skills learned in this class will help students succeed in technical interviews as well as in major programming competitions such as the International Collegiate Programming Contest and the Google Code Jam. Topics will include fast problem analysis, algorithm design and implementation, and team programming.

Prerequisites Fluency in a modern computer programming language, and knowledge of basic algorithms and data structures. May be taken concurrently with CSE 241.

2 Logistics

Staff This course is being primarily taught by Shane Carr, with Joey Woodson as TA and Ron Cytron as faculty advisor.

Textbook There is no required textbook for the class, but the material will be heavilly drawn from *Competitive Programming 3* by Steven Halim and *Cracking the Coding Interview* by Gayle McDowell. Both books are inexpensive and readily available online.

Meeting Time Class meetings will be held 1:00-2:30 PM on Fridays in Urbauer 214. Attendance will be taken; see the "Grading" section below.

Additional Events Students will also need to participate in at least one programming contest outside of class. Some of these contests have set dates and times, while others can be done at a time and place convenient for the student. More details are provided in the "Programming Contests" section.

3 Grading

This is a pass-fail course. In order to earn a "Pass" grade, the student will need to earn 65 points of 100 possible points through the course of the semester. Points are available in the following categories:

1. **Attending Lecture (26 points):** There are 13 lectures during the semester. Attending lecture and asking at least one question is worth 2 points; attending lecture but having poor participation is worth 1 point; and unexcused absences are worth 0 points.

   Excused Absences: Requests for excused absences must be sent to the instructor in advance of the missed class period and will be handled on a case-by-case basis.

   Extra Credit Presentations: On most weeks when there is a programming assignment due, I will ask for a volunteer from the class to present their solution to the problem. This is intended to be a way for the student to practice explaining their code in front of an audience. The student will earn 2 bonus points each time they present.

2. **Weekly Programming Challenges (34 points):** A programming challenge will be posted each week in our course's HackerRank account to supplement the material covered in lecture that week. The challenges will be worth either 2 points or 3 points depending on their difficulty. The challenges will be auto-graded and must be completed before the following lecture.

3. **Real-World Programming Contests (40 points):** The remaining points come from the out-of-class programming contests, described in the next section.

These points are structured in such a way that there are multiple ways a student can earn a passing grade. If the student comes to class and participate every day, does the weekly programming assignments, and participates in one real-world programming contest, they will earn enough points to pass. Likewise, if the student participates in most of the real-world programming contests, they will be able to pass the class without completing all of the weekly programming assignments.

There will additionally be extra credit provided to students who perform exceptionally well in the real-world programming contests (e.g., placing as a semi-finalist or qualifying for ICPC regionals).

Collaboration: Students are expected to abide by the standard collaboration policy for computer science courses at Washington University. This policy can be found on the CSE 241 web site at the following URL. `http://classes.engineering.wustl.edu/cse241/handouts/collaboration.pdf`

4 Additional Programming Contests

There will be at least 40 points available from additional programming contests, plus more points available for extra credit. If there is a programming contest you would like to participate in but which is not on this list, contact the instructor at least a week in advance to see if it can count for course credit.

1. **ICPC Qualifiers (15 points):** This contest will take place on Saturday, October 3, 2015. It is a simulation of the full 5-hour ICPC contest that takes place in November. In order to earn credit, the student must stay for the entire duration of the contest and solve at least one problem.

2. **Mock Google Code Jam (15 points):** We will host a mock Google Code Jam contest on Saturday, November 21. In order to earn credit, the student must stay for the entire duration of the contest and solve at least one problem.

3. **Mock Technical Interviews (5 points):** There will be a mock technical interview session later in the semester (exact date/time TBA). The student will earn 5 points for participating in this event.

4. **CodeSprint (5 points + up to 10 points extra credit):** This is an online contest held over three weekends in September. The student will earn 5 points for participating in the first round and solving at least one problem. The student will earn 5 bonus points if the student qualifies for and participate in the semi-finals round, and another 5 bonus points for the finals round.

5. **ICPC Regionals (up to 10 points extra credit):** If the student scores well in the ICPC Qualifiers, the student will earn the opportunity to represent WashU in the ICPC Regionals in November. The student will earn 5 bonus points for participating in the ICPC Regionals, plus another 5 bonus points for being on a team ranking in the top 20 in the region.

5 Class Schedule

The following schedule is a rough outline and is subject to change. The topics covered during the 13 lectures will include:

1. Introduction

2. Functional Programming

3. Skills for Technical Interviews

4. In-class technical interview session (week of the career fair)

5. Dynamic Programming

6. Binary Mathematics
7. Graph Theory 1
8. Graph Theory 2
9. Computational Geometry
10. Probability and Combinatorics
11. Number Theory
12. Problem Analysis
13. End-of-semester activity