CSCI 2190 – Fall 2015 Handout: syllabus
Computational Problem Solving
Michael H. Goldwasser
Saint Louis University Tuesday, 25 August 2015

CSCI 2190: Computational Problem Solving

1 Overview

Each year the Association for Computing Machinery (ACM) sponsors the International Collegiate Programming Contest (ICPC). Last year's contest involved over 38,000 students from over 2,534 universities in 101 countries. In the Fall, teams compete in one of 459 different regional competitions; the top 128 teams in the world advance to the World Finals in the Spring. (The 2015 World Finals was held in Marrakesh, Morocco; the 2016 World Finals will be held in Phuket, Thailand.)

Saint Louis University competes as part of the Mid-Central Region, which includes Missouri, Illinois, Kentucky, Tennessee, Arkansas, and parts of Indiana. This year's regional contest will be held on **Saturday, 7 November 2015**; we will compete at Webster University.

2 Contest Logistics

Students compete in teams of three; SLU typically enters two such teams to the regional competition. Each team shares a single computer and is presented with 6–9 programming challenges to solve during a five-hour time period. The available programming languages for the contest are C++ and Java.

Each challenge is structured such that there is a prescribed input format and a well-defined expected output. Teams are given sample inputs and outputs, but the judges retain an additional set of test cases that are not known to the competitors. During the contest, teams may submit source code to solve a particular problem. The solution will be deemed correct if they produce precisely the correct output on the entire judge's suite of tests; incorrect submissions are returned to the team. The goal is to solve as many problems as possible during the five-hour time period.

3 Team Website

`http://cs.slu.edu/~goldwasser/icpc`

4 CSCI 2190 Logistics

CSCI 2190 is a 1-credit hour course, offered only under the P/NP grading scheme. Intended primarily to train students for the International Collegiate Programming Contest (ICPC), this course covers data structures, algorithms, and programming techniques that apply to typical programming challenges.

Meetings of the course are intentionally coordinated with weekly practices for the programming competition. With that said, **students who wish to compete in the programming competition need not register for CSCI 2190, and students may register and receive credit for CSCI 2190 even if they do not compete in the programming contest.**

4.1 Prerequisites

Given the nature of typical problems, students should have previously completed CSCI 2100 (Data Structures) or equivalent.

4.2 Instructor

Instructor:	Dr. Michael Goldwasser
Email:	`goldwamh@slu.edu`
Web:	`http://cs.slu.edu/~goldwasser/`
Office:	Ritter Hall 335
Phone:	(314) 977-7039
Office hours:	Tuesdays 3:00–4:00pm
	Wednesdays 2:00–3:00pm
	Fridays 12:00–1:00pm
	or by appointment

4.3 Class Meetings

The group will meet once a week for two hours, from the beginning of the semester until a final postmortem session just after the contest occurs during the first week of November. Meetings will be held on Tuesdays, 4:10–6:00pm in Ritter Hall 117.

4.4 Textbook

There is not required textbook for this class, and we will distribute a lengthy document that serves as a "crib sheet" for our team on the day of the contest. Two very valuable readings for contest participants is the following:

Title:	*Competitive Programming, Third Edition*
Authors:	Steven Halim and Felix Halim
Publisher:	Lulu, 2013
ISIN:	B00FG8MNN8

Title:	*Programming Challenges: The Programming Contest Training Manual*
Authors:	Steven S. Skiena and Miguel A. Revilla
Publisher:	Springer, 2003
ISBN:	978-0387001630

4.5 Requirements for a Passing Grade

Students will receive a passing grade in the course by attending at least 80% of the class meetings, and by successfully completing at least 5 past contest problems of varying difficulty as practice.

4.6 Learning Outcomes

At the completion of this course, students will be able to:

- Read a problem statement and recognize the necessary algorithmic approach to solve the problem.
- Convert a high-level approach into complete code for solving the problem in either C++ or Java.

4.7 Topical Outline

- Parsing Input
- Generating Output
- Sorting/Searching Algorithms
- String Manipulation
- Brute-Force Enumeration
- Search Spaces
- Mathematical Fundamentals
- Grid and Lattice Problems
- Dynamic Programming
- Graph Algorithms

4.8 Academic Integrity

This course is extremely collaborative by nature, given the need for developing effective teamwork. The only expectation is that solutions to practice problems that are submitted for course credit be genuinely authored by that student (or team of students).

4.9 Supporting Student Success

In recognition that people learn in a variety of ways and that learning is influenced by multiple factors (e.g., prior experience, study skills, learning disability), resources to support student success are available on campus. Students who think they might benefit from these resources can find out more about:

- Course-level support in the remainder of this section or by asking the instructor
- University-level support (e.g., tutoring/writing services, Disability Services) by visiting the Student Success Center (BSC 331) or by going to `www.slu.edu/success`.

Students with a documented disability who wish to request academic accommodations are encouraged to contact Disability Services at 314-977-3484 or visit the Student Success Center. Confidentiality will be observed in all inquiries.