Problem-Solving Skills for the Programming Interview - Syllabus

Code: CCOM 4995
Credit/Hours: 3 credits / 3 hours of conference and practice
Prerequisites: CCOM 3034
Corequisites: None
Semester: Spring 2019

Description:

Solving real-world computational problems requires a good working knowledge of data structures and algorithms. Questions that appear in programming interviews can have deceivingly simple statements. However the first solution is almost never the best. Thinking through the obstacles in order to obtain a better solution is a discovery process that leads to a deeper understanding about what certain algorithms and data structures do, and why people studied them in the first place. This course aims to help the student be more prepared for a programming interview by acquiring a working knowledge of data structures and algorithms through programming interview questions, along with a dialectical in-class demonstration of problem-solving methodologies. (Taken from Online Reference [1])

Objectives [from Problem Solving Course 3 by Google and CAHSI]

At the end of this course the student will be able to:

1. Describe problem-solving strategies/approaches, e.g. IDEAL, Polya's Problem Solving Techniques and Duke's 7 steps. In particular, he/she will be able to (a) describe the role of abstraction in analyzing a problem description, (b) describe the difference between clarifying and probing questions, and (c) understand the importance of basic data structure and algorithm knowledge.

2. Apply problem-solving strategies to coding interview problems, including abstraction, question generation (clarifying and probing), data collection and analysis, problem decomposition, and pattern generalization.

3. Communicate (oral and written) solutions to technical/coding problems.

4. Solve technical/coding problems with redundant, incomplete, and inconsistent specifications.

5. Evaluate correctness and quality of different solutions to technical/coding problems using metrics such as efficiency, correctness, and coverage.

6. Work efficiently in groups to provide solutions to problems, defend decisions and accept/provide constructive critique of solutions presented by other groups.

Text Book

McDowell, Gayle Laakmann. Cracking the Coding Interview: 189 Programming Questions and Solutions. CareerCup, LLC, 2016.

Content and time distribution

Topic	**Hours**
Problem Solving Strategies and Approaches	6
A review of Complexity	1.5
Problem solving techniques for various subdomains:	
-- Data structures	4.5
-- Bit manipulation	3
-- Math and Logic Puzzles	6
-- Object Oriented Design	6
-- Recursion and Dynamic Programming	6
-- System Design and Scalability	6
Testing (in context of technical interviews)	3
Problems dealing with programming language specifics	3
Total	45

Evaluation strategies

Midterm Exam	20%
Final Exam	20%
Class participation	20%

In class and homework programming assignments	40%

Asistencia

La asistencia diaria y puntual a clase es requerida, tal como lo indica el Reglamento Universitario. Una ausencia, por la razón que sea, excusada o no, meritoria o no, nunca exime al estudiante de la responsabilidad de mantenerse al día en el curso y de hacer los trabajos asignados. Por esta razón el estudiante debe comunicarse con algún compañero de clase para estudiar el material cubierto en la clase a la que faltó. El profesor podrá indicarle generalmente sobre los temas cubiertos, pero no dará clases individuales de reposición.

Derechos de Estudiantes con Impedimentos

La Universidad de Puerto Rico cumple con todas las leyes federales y estatales, y reglamentos concernientes a discriminación, incluyendo "The American Disabilities Act" (Ley ADA) y Ley 51 del Estado Libre Asociado de Puerto Rico. Como la meta del curso es preparar a los estudiantes al maratón de programación ACM/ICPC, los ejercicios en clase y el examen en blanco serán efectuados con los tiempos recomendados por ACM/ICPC, para que todos los estudiantes sean sometidos a pruebas que simulen lo mejor posible el ambiente de competencia.

Alternate teaching methods:

La Certificación Núm 112 (2014-2015) de la Junta de Gobierno define un curso presencial como un curso en el cual 75% o más de las horas de instrucción requieren la presencia física del estudiante y el profesor en el salón de clases. Esto quiere decir que 25% de un curso presencial, pudiera ofrecerse sin requerir la presencia física de los estudiantes y el profesor en el salón de clases. En caso de ser necesario, este curso podrá completar hasta 25% de las horas contacto (11.25 horas) de forma no presencial por métodos alternos como por ejemplo: Videoconferencias, módulos instruccionales, foros de discusión y cibercharlas entre otros. De ser así, se modificará el calendario/temario para incluir los temas que serán cubiertos por métodos alternos.

References

1. Mongan, John, Noah Suojanen Kindler, and Eric Giguère. Programming interviews exposed: secrets to landing your next job. John Wiley & Sons, 2012.
2. Robert Sedgewick. Algorithms in C++, Third Edition. Addison Wesley. 1998.
3. T.H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein Introduction to Algorithms (2nd Edition), The MIT Press, Mc-Graw-Hill (2001)
4. http://www.cplusplus.com/doc/language/tutorial/structures.html

5. A.V. Aho, J.E. Hopcroft, and J.D. Ullman: Data Structures and Algorithms, Addison-Wesley (1983)
6. Donald E. Knuth: The Art of Computer Programming, Addison-Wesley (1997)

Online References

1. I. Koutis, Fun with Programming Interview Questions - Course Syllabus http://ccom.uprrp.edu/~ikoutis/classes/fun_15/Fun_15.htm
2. Project Euler. https://projecteuler.net/
3. LeetCode. https://leetcode.com/
4. Pat Morin. Open Data Structures. Licensed under a Creative Commons License. Available at: http://www.aupress.ca/books/120226/ebook/03_Morin_2013-Open_Data_Structures.pdf
5. Programming Abstractions (Stanford University) - http://www.stanford.edu/class/cs106b/
6. Notes for the Sedgewick's Algorithms book: http://algs4.cs.princeton.edu/home/