Rutgers University, Department of Electrical and Computer Engineering
ABET COURSE SYLLABUS
COURSE: 14:332:351

Course Catalog Description: 14:332:351 - Programming Methodology II (3)
In-depth analysis of analysis and design of software and programming methodologies. Scalable design approaches. In-depth analysis of algorithms using object oriented techniques. Comparative algorithm analysis, in-depth sorting algorithms, graphs, object-oriented design. Programming languages include C++.

Pre-Requisite Courses: 14:332:252 or the equivalent

Co-Requisite Courses: None

Pre-Requisite by Topic:
1. Knowledge of C++ language.
2. Stacks, queues, linked lists.
3. Sorting algorithms.

Textbook & Materials: F. Carrano, *Data Abstraction & Problem Solving with C++, 6th Ed*, Prentice Hall, 2014.

References: None

Overall Educational Objective: To develop in depth skills in efficient design of algorithms.

Course Learning Outcomes: A student who successfully fulfills the course requirements will have demonstrated:

1. advanced programming skills
2. advanced knowledge of algorithms

How Course Outcomes are Assessed:
HW Problems (15 %)
Two Mid-Term Exams (50 %)
Final Exam (35 %)

N = none S = Supportive H = highly related

Outcome	Level	Proficiency assessed by
(a) an ability to apply knowledge of Mathematics, science, and engineering	H	Programming assignments, Exams
(b) an ability to design and conduct experiments and interpret data	N	
(c) an ability to design a system, component or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	N	
(d) an ability to function as part of a multi-disciplinary team	N	
(e) an ability to identify, formulate, and solve ECE problems	H	Programming assignments, Exams
(f) an understanding of professional and ethical responsibility	N	
(g) an ability to communicate in written and oral form	S	Programming assignments, Exams
(h) the broad education necessary to understand the impact of electrical and computer engineering solutions in a global, economic, environmental, and societal context	N	
(i) a recognition of the need for, and an ability to engage in life-long learning	S	Programming assignments
(j) a knowledge of contemporary issues	N	
(k) an ability to use the techniques, skills, and modern engineering tools necessary for electrical and computer engineering practice	H	Programming assignments, Exams

Basic disciplines in Electrical Engineering	N	
Depth in Electrical Engineering	N	
Basic disciplines in Computer Engineering	H	Programming assignments, Exams
Depth in Computer Engineering	S	Programming assignments, final project, Exams
Laboratory equipment and software tools	H	Programming environments
Variety of instruction formats	S	Lecture, in-lab lectures, online discussion, online submission, office hour discussions

Topics Covered week by week:

Weeks 1 & 2: Review of Data Structures portion of PM-I, stacks, queues, linked lists, sorting algorithms
Week 3: Basics of object oriented programming (C++). Advanced problems using recursion.
Week 4: Analysis and design of Software, e.g. UML. Characteristics of "good design".
Week 5: Standard conversion under derivation, virtual functions, virtual base classes, OO design.
Week 6: Algorithm Analysis, Big-Oh notation, Solution of Recurrence Equations
Week 7: Multiway Search Trees, Top Down Trees, Traversal and Insertion in Top Down Trees
Week 8: MIDTERM EXAM; B-Trees, Search Traversal and Insertion
Week 9: Implementation of algorithms for B-Tree
Week 10: Efficiency of B-Tree and Top Down Trees; B+ Trees and algorithms to implement them
Week 11: Graphs, Adjacency Matrix Representation, Transitive Closure; Transitive Closure using Warshall's Algorithm
Week 12: Shortest Path Algorithm, Adjacency List representation of Graph, Network Flow Problem and the algorithm to compute the optimal flow
Week 13: Spanning Forests of Graph, Graph Traversal, Depth First Traversal, Breadth First Traversal
Week 14: Minimum spanning Trees; Introduction to Java
Week 15: Final Examination

Computer Usage: Use of C++ and Java to implement advanced algorithms.

Laboratory Experiences: Implementation of algorithms in C++ and Java.

Design Experiences: Moderate design experience in constructing C++ programs, simple design experience construction Java programs.

Independent Learning Experiences: Programming assignments (homework), final lab project

Contribution to the Professional Component:
(a) College-level mathematics and basic sciences: 0.25 credit hours
(b) Engineering Topics (Science and/or Design): 2.75 credit hours
(c) General Education: 0 credit hours
Total credits: 3
Prepared by: Shantenu Jha
Date: August 2015