

EECS 268 Programming II

General Information – Fall 2012

Class Information: As below:

Lectures:

Room: Mallot 2001, Time: MWF 2:00PM – 2:50PM

Lab:

Room: Eaton 1005D, Time: M 3:00PM – 4:50PM,

Room: Eaton 1005D, Time: T 11:00AM – 12:50PM

Room: Eaton 1005D, Time: W 3:00PM – 4:50PM

Room: Eaton 1005D, Time: F 9:00AM – 10:50AM

Prerequisites: EECS168/169 (Programming I),

Instructor: Prasad Kulkarni

Office: 2030 Eaton (Ph: 785-864-8819)

Office Hours: MW 3:30PM – 5:00PM or by appointment

Email: prasadm@ku.edu

Other Location: 137 Nichols (Ph: 785-864-7322)

Teaching Assistants: We have two GTAs for this class:

Joseph St.Amand Office: Eaton 3015 Office Hours: TR 9:00AM – 10:00AM Email: josephstamand@gmail.com Labs: MW 3:00PM – 4:50PM	Adam Van Horn Office: Eaton 3015 Office Hours: T 10-11AM, F 1-2PM Email: bog411@gmail.com Labs: T 11:00AM – 12:50PM and F 9:00AM – 10:50AM
--	---

Text: Data Abstraction & Problem Solving With C++ (Walls and Mirrors) by Frank Carrano – Fifth edition (ISBN: 0-321-43332-7).

Catalog Listing: This course continues developing problem solving techniques by focusing on the imperative and object-oriented styles using Abstract Data Types. Basic data structures such as queues, stacks, trees, and graphs will be covered. Recursion. Basic notions of algorithmic efficiency and performance analysis in the context of sorting algorithms. Basic Object-Oriented techniques. An associated laboratory will develop projects reinforcing the lecture material. Three class periods and one laboratory period per week.

Class Homepage: The class home page is at <http://www.ittc.ku.edu/~kulkarni/teaching/EECS268/>. The page will contain a variety of information, which will include the syllabus, schedule, slides, and assignments.

Course Objectives: The goal of this class is to continue the development of programming and logical problem solving skills. In particular, we will learn:

1. **Data Structures and Their Use During Problem Solving** – We will study common methods for storing and manipulating data to effectively solve complex problems. Important abstract data structures, including lists, stacks, queues, trees, graphs, and hash-tables will be introduced. Data structures will be used during problem solving.

- 2. Object Oriented Programming in C++** – We will study basic OOP concepts, including data abstraction, data hiding, overloading, polymorphism, etc.
- 3. Programming** – A major emphasis is on getting comfortable with logical problem solving and programming to implement an efficient solution. Software engineering principles for coding, debugging, and maintenance will be discussed and employed.

Class Programming Environment: All program development and programming-related activity will be done using the C++ language as supported on the EECS departmental Linux systems. It is your responsibility to make sure the code you submit compiles and runs correctly in the Linux environment used for the course.

Grading: Grades will be based on your scores over three exams (15% – Midterm-1, 15% – Midterm 2, 25% – Final, **Total: 55%**), take-home quizzes (**Total: 10%**), and programming projects/labs (**Total: 35%**). Keep all graded material to provide evidence of grades. The final exam may be comprehensive.

Exam Schedule: Here is the tentative Exam schedule. Midterm-1 on Wednesday 9/19, Midterm-2 on Friday 10/19, and Final on Wednesday 12/12 (1:30 - 4:00pm).

Attendance and Punctuality: Roll is not taken, but you are responsible for all material presented in class. You must come to the class and labs properly prepared to get the most knowledge out of the class. This includes understanding the previously covered material and reading the lecture slides/lab assignment before coming to class/lab.

Exams and lab due dates will be scheduled in advance. A grade of zero will be recorded for missed exams, late lab projects, and late quizzes unless prior arrangements are made. Lab projects and take-home quizzes turned in after the due date, but by the beginning of the next scheduled class/lab will be penalized 25%. Projects/quizzes will not be accepted that are more than one class/lab period late.

Cheating: Students are encouraged to discuss programs in general and to help one another find bugs in existing programs. Copying another's code or writing code for someone else is cheating. Keep listings to provide evidence of creative development. Please review your student handbook for additional details on what constitutes academic misconduct.

Please advise the instructor of this class at your earliest convenience (minimum of five working days) if you have a disability that will require a reasonable accommodation for any of the activities in the course schedule.