UTSA
Department of Electrical and Computer Engineering
EE 5103 – Engineering Programming
Syllabus – Spring 2020

Part A - Course Outline

Course Description:

EE 5103. Engineering Programming. (3-0) 3 Credit Hours.

Prerequisite: Graduate standing or consent of instructor.

Object oriented programming for engineering design problems using C++; software development for mathematical modeling and simulation of hardware systems; extraction and reporting (e.g., text processing) using scripting languages such as Perl; and individual class projects.

Prerequisites:

Graduate standing or consent of instructor.

Course Objectives:

1. Demonstrate basic knowledge in core C++ concepts
2. Demonstrate basic ability to design and model classes/objects in C++
3. Demonstrate advanced knowledge in object oriented programming in C++
4. Demonstrate basic knowledge of using native C++ libraries.

Evaluation Methods:

1. Exams
2. Assignments
3. Project

Performance Criteria:

Course objectives 1 through 4 will be evaluated using evaluation methods [1 - 3]

Course Content:

Engineering Science: 2 credits (67%)
Engineering Design: 1 credit (33%)

Class/Laboratory Schedule:

2 hours and 30 minutes of lectures per week.

Course coordinator:

Ram Krishnan – Associate Professor of Electrical and Computer Engineering

Part B – General Course Information and Policies

Instructor:

Ram Krishnan (http://engineering.utsa.edu/rkrishnan/)
Microsoft President's Endowed Associate Professor
Department of Electrical and Computer Engineering
University of Texas at San Antonio
Email: Ram.Krishnan@utsa.edu

Lecture hours:

Tuesdays and Thursdays 2:30 PM – 3:45 PM @ EB 2.04.04

Office hours:

Tuesdays and Thursdays 10:00 AM – 11:15 PM. Instructor's office is at BSE 1.518.

Course website:

http://engineering.utsa.edu/rkrishnan/teaching/ee-5103-engineering-programming/

The above website will be used for general info dissemination. Content (assignments, etc.) for this course will be managed through Blackboard.

Recommended Textbook:

"C++ Primer, Fifth Edition", by Stanley B. Lippman, Josee Lajoie, and Barbara E. Moo

Topics:

1. The Basics
   a. Variables and Basic Types
   b. Strings, Vectors and Arrays
   c. Expressions
   d. Statements
   e. Functions
   f. Classes
2. The C++ Library
   a. IO Library
   b. Sequential Containers
   c. Generic Algorithms
   d. Associative Containers
   e. Dynamic Memory
3. Tools for Class Authors
   a. Copy Control
   b. Overloaded Operations and Conversions

    c. Object Oriented Programming
    d. Templates and Generic Programming
4. Advanced Topics
    a. Specialized Library Facilities
    b. Tools for Large Programs

Evaluation methods:

1. Two Exams – 50% (20% mid-term exam + 30% final exam)
2. Assignments – 30%
3. Project – 20%

Grading:

A letter grade will be determined based on the nature of students' course performance curve.

Attendance:

No penalties will be incurred for absences during regular class hours. However, it is your responsibility to talk to your classmates and keep abreast of topics covered, announcements and assignments given during missed classes.

Late submission policy for assignments and project:

Late submission is not allowed.

Exam policy:

Exams will be held in-class, closed-book and closed-notes. To be fair to all students, there will be no makeup exams. No credit will be given for a missed exam except under extenuating circumstances such as an unexpected major health issue.

Course evaluation:

Each student completing this course is highly encouraged to evaluate the course toward the end of the semester. The evaluation is used for 2 major purposes: (1) The instructor strongly takes the feedback into account to improve his teaching in the future, and (2) The university utilizes the feedback as one measure to evaluate instructor effectiveness. To encourage student participation, the instructor offers a 1% extra-credit for each student who completes his/her course evaluation.

Counseling services, student code of conduct and scholastic dishonesty, etc.:
Please visit this webpage: **http://utsa.edu/syllabus**

Tentative course schedule:

Please take a look at the Spring academic calendar and the Spring final exam schedules in UTSA ASAP. This is a tentative schedule of lecture topics. We will likely calibrate as we move along. In particular, the Mid-Term exam date is tentative (± 1 week).

14 lectures before Mid-Term

Expected date of Mid-Term Exam: Mar 17 (± 1 week)

Lecture #	Date	Topics Covered
1	Jan 21	Course overview.
2	Jan 23	Course overview, Hello world, variables and basic data types
3	Jan 28	Strings, vectors and arrays
4	Jan 30	Expressions
5	Feb 04	Statements
6	Feb 06	Functions
7	Feb 11	Classes
8	Feb 13	Classes
9	Feb 18	IO library
10	Feb 20	IO library
11	Feb 25	Sequential containers
12	Feb 27	Sequential containers
13	Mar 03	Generic algorithms
14	Mar 05	Generic algorithms and mid-term review
	Mar 10	**Spring break**
	Mar 12	
	Mar 17	**Mid-term exam (instructor on travel)**

15 lectures before Final

Expected Project deadline: 05/05
Final Exam: May 07 @ 12:30 PM

Lecture #	Date	Topics Covered
16	Mar 19	Associative containers
17	Mar 24	Associative containers
18	Mar 26	Dynamic memory
19	Mar 31	Dynamic memory
20	Apr 02	Class copy control (Expected date of project announcement)
21	Apr 07	Overloaded operations
22	Apr 09	Overloaded operations/conversions
23	Apr 14	Object-oriented programming
24	Apr 16	Object-oriented programming

25	Apr 21	Templates and generic programming
26	Apr 23	Templates and generic programming
27	Apr 28	No face to face class: attend Tech Symposium
28	Apr 30	Specialized library functions, Tools for large programs
29	May 05	Tools for large programs, Final exam review
	May 07	**Final exam @ 12:30 PM**
	May 18	Course grade due @ 2:00 PM