Introduction to Computer Programming

ASE 301 / COE 301
Unique Number: 13590
Fall 2017

Instructor:	Amir Shahmoradi
office:	POB 6.326
e-mail:	amir@ices.utexas.edu
office hours:	TBD

Lecture meeting times:	MWF 9 - 10 am
Lecture meeting place:	UTC 4.110

Teaching Assistants:	Vivek Varier	Travis Driver	TBD
office:	TBD	TBD	TBD
e-mail:	viv.varier@utexas.edu	travis.driver7@utexas.edu	TBD
office hours:	TBD	TBD	TBD

COURSE OBJECTIVES / ACADEMIC LEARNING GOALS

The primary objective of this course is to learn basic computer programming concepts and apply them to engineering computations. By the end of the course, you should have a good understanding of programming practices and be able to analyze engineering/mathematical problems and develop computational solutions for them, potentially collaboratively within a team. We will achieve this by learning how to program in MATLAB, Fortran, and C++. No prerequisites are required, although some knowledge of calculus and linear algebra is useful.

COURSE SCHEDULE

The following is a tentative outline of the topics to be covered.

- Version Control Systems (VCS): Principles of professional project management and collaborative programming with the use of Git.
- Programming History – operating systems – roundoff and truncation errors – number systems.
- Principles of Programming using MATLAB: The MATLAB development environment; syntax rules; variables and data types; conditionals; looping; input/output; functions; M files; functions; matrix operations; plotting; symbolic calculations; file processing; etc.
- Compiled languages: Fortran and C++. general syntax rules; variable types; conditionals and looping; functions and subroutines; arrays; input/output;

COURSE TEXTBOOKS

No textbook is required for this course. Online class lecture notes will be used as reference. However, the following is a list of textbooks for those who are interested to self-educate themselves or go beyond class syllabus.

MATLAB:

1. Chapman, 2016, MATLAB Programming for Engineers.
2. Attaway, 2014, Matlab: A Practical Introduction to Programming.
3. King, 2017, MATLAB Programming for Biomedical Engineers and Scientists.
4. Van Loan, 2010, Insight through Computing, A MATLAB Introduction.
5. Driscoll, 2009, Learning MATLAB.
6. Higham, 2005, Matlab Guide.
7. Moler, 2004, Numerical Computing with Matlab.

Fortran:

1. Metcalf, 2011, Modern Fortran Explained.
2. Chivers, 2012, Introduction to Programming with Fortran.
3. Chapman, 2017, Fortran for Scientists and Engineers.
4. Clerman, 2012, Modern Fortran: Style and Usage.

C++:

1. Gottschling, 2016, Discovering Modern C++: An Intensive Course for Scientists.
2. Horstmann, 2011, C++ For Everyone.

COURSE LOGISTICS

Grading:
Homework: 25% (Each assignment might not be weighted the same)
Random Weekly Quizzes: 25%
Midterm Exam: 25%
Final Exam: 25%

Homework Policy:
There will be approximately one homework per lecture. Assignments will be due before lecture begins, and should be added to an online repository determined by the instructor. No late assignment will be accepted. No exceptions to the homework policy will be made without prior instructor approval.

Examinations:
The midterm exam will cover the topic from the beginning of the semester to the date of exam. The final exam will be more focused on the topics covered after midterm exam.

Attendance:
Regular attendance is expected. Any absence requires prior approval from the instructor, or compelling evidence of illness or an official letter from the university administration. Student attendance will be randomly checked.

Scholastic dishonesty: All students are responsible for upholding the University rules on scholastic dishonesty. Students who violate University rules on scholastic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the University. Since such dishonesty harms the individual, all students, and the integrity of the University, policies on scholastic dishonesty will be strictly enforced. For further information,

visit the Student Judicial Services web site http://deanofstudents.utexas.edu/sjs/, and the General Information Catalog information http://catalog.utexas.edu/general-information/ .

Other matters: The University of Texas at Austin provides, upon request, appropriate academic adjustments for qualified students with disabilities. Any student with a documented disability (physical or cognitive) who requires academic accommodations should contact the Services for Students with Disabilities area of the Division of Diversity and Community Engagement at 471-6259 as soon as possible to request an official letter outlining authorized accommodations. For more information, contact that office at 471-6259, Video Phone 410-6644, or http://www.utexas.edu/diversity/ddce/ssd.

Your Expectations:

For the Fall 2017 offering of this course, we will cover the principles of computer programming using MATLAB programming language, as well as using important modern compiled languages that are widely used in scientific computation: Fortran and C++. Specifically, upon completion of this course students will be familiar with,

- programming paradigms,
- principles of software maintenance and collaborative project development,
- differences between compiled and interpreted programming languages,
- how to use MATLAB as a simple calculator,
- how to use MATLAB as an advanced scientific computation and graphics toolbox,
- how to compile and write scientific code in modern compiled languages such as Fortran and C++,
- how to formulate cast a scientific problem in the form of a computational programming algorithm.

Course Schedule:

The following is the tentative schedule of topics to be covered, and will be continuously updated. The exam dates are final.

Wed Aug 30	Student-professor connection day; course outline
Fri Sep 01	Programming history and paradigms
Mon Sep 04	NO CLASS: LABOR DAY HOLIDAY
Wed Sep 06	VCS: setting up your Git Version Control System and Github account
Fri Sep 08	VCS: advanced topics in Version Control System;
Mon Sep 11	MATLAB: installation and setup for beginners –HW 1
Wed Sep 13	MATLAB: values, variables, and types I
Fri Sep 15	MATLAB: values, variables, and types II
Mon Sep 18	MATLAB: branching and control statements – HW 2
Wed Sep 20	MATLAB: functions I
Fri Sep 22	MATLAB: functions II
Mon Sep 25	MATLAB: functions III – HW 3
Wed Sep 27	MATLAB: free discussion
Fri Sep 29	MATLAB: loops, array computing, and vectorization
Mon Oct 02	MATLAB: input/output (I/O) I – HW 4
Wed Oct 04	MATLAB: input/output (I/O) II, Exception Handling
Fri Oct 06	MATLAB: plotting I
Mon Oct 09	MATLAB: plotting II – HW 5
Wed Oct 11	MATLAB: root finding and optimization
Fri Oct 13	MATLAB: differential equations
Mon Oct 16	MATLAB: deterministic integration – HW 6
Wed Oct 18	MATLAB: Monte Carlo integration
Fri Oct 20	MATLAB: statistical analysis
Mon Oct 23	MATLAB: OOP I – HW 7
Wed Oct 25	MATLAB: OOP II
Fri Oct 27	MATLAB: OOP III / semester final project description
Mon Oct 30	NO LECTURE: MID-TERM EXAM
Wed Nov 01	C++/Fortran: compiler installation and setup
Fri Nov 03	C++/Fortran: values, variables, and types I
Mon Nov 06	C++/Fortran: values, variables, and types II – HW 8
Wed Nov 08	C++/Fortran: branching statements I
Fri Nov 10	C++/Fortran: branching statements II
Mon Nov 13	C++/Fortran: loops I – HW 9
Wed Nov 15	C++/Fortran: loops II
Fri Nov 17	C++/Fortran: array computing, and vectorization I
Mon Nov 20	C++/Fortran: array computing, and vectorization II – HW 10
Wed Nov 22	NO CLASS: THANKSGIVING HOLIDAYS
Fri Nov 24	NO CLASS: THANKSGIVING HOLIDAYS
Mon Nov 27	C++/Fortran: dynamic memory allocation – HW 11
Wed Nov 29	C++/Fortran: pointers
Fri Dec 01	C++/Fortran: functions and subroutines I
Mon Dec 04	C++/Fortran: functions and subroutines II – HW 12
Wed Dec 06	C++/Fortran: input/output (IO)
Fri Dec 08	C++/Fortran: standard libraries / intrinsic procedures
Mon Dec 11	ICP wrap-up: course summary, advice for your future career – HW 13