ECTS COURSE INFORMATION FORM

Faculty	**Faculty of Engineering**	
Program	**B.Sc. in Civil Engineering**	**Elective**
	B.Sc. in Computer Engineering	**Required**
	B.Sc. in Electrical-Electronics Engineering	**Elective**
	B.Sc. in Industrial Engineering	**Elective**
	B.Sc. in Mechanical Engineering	**Elective**

Course Code	COMP 205			
Course Title in English	Systems Programming			
Course Title in Turkish	Sistem Programlama			
Language of Instruction	English			
Type of Course	Flipped Classroom/Lecture/Exercise			
Level of Course	Undergraduate			
Course Category (by % of Content)	Basic Science	Basic Engineering	Engineering Design	General Education
	10	30	60	
Semester Offered	Fall			
Contact Hours per Week	Lecture: 3 hours	Recitation: -	Lab:2 hours	Other:-
Estimated Student Workload	148 hours per semester			
Number of Credits	6 ECTS			
Grading Mode	Standard Letter Grade			
Pre-requisites	COMP 109			
Expected Prior Knowledge	Basic programming knowledge			
Co-requisites	None			
Registration Restrictions	Only Undergraduate Students			
Overall Educational Objective	To learn fundamentals of systems programming concepts and construct basic system software using C programming language on UNIX-based environment.			
Course Description	This course covers the fundamentals of systems programming concepts such as machine-level representation of programs, program linking, and system-level I/O. Application of these concepts are realized in C programming language on UNIX-based environment.			
Course Description in Turkish	Bu ders sistem programlama kavramlarının temellerini içerir (makine seviyesi program gösterimi, program birleştirme ve sistem seviyesinde girdi/çıktı). Kavramların uygulaması, UNIX tabanlı sistemlerde C programlama dili kullanılarak yapılır.			
Course Learning Outcomes and Competences	Upon successful completion of the course, the learner is expected to: 1. comprehend basic systems programming concepts; 2. use Unix-based environment; 3. use C programming language to design algorithms; 4. design basic systems software to solve simple engineering problems.			

Relationship of the Course with the Student Outcomes	Level	Learning Outcome(s)	Assessed by
Student Outcomes	N=None S=Supportive H=High		Exam, Project, HW, Experiment, Presentation, etc.
(1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics	S	1	Exam, Lab Practice
(2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors	H	2,3,4	Exam, Lab Practice, Project
(3) an ability to communicate effectively with a range of audiences			
(4) an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts			
(5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives			
(6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions			
(7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies			

Prepared by and Date	Asst. Prof. Şeniz Demir / June 2019	
Semester	Fall 2019-2020	
Name of Instructor	Asst. Prof. Şeniz Demir	
Course Contents	Week	Topic
	1.	Introduction to Systems Programming (shell, compiler, assembler, linker, loader)
	2.	Unix Environment: Introduction to UNIX-based systems
	3.	Unix Environment: Command-line instructions
	4.	Shell Programming
	5.	Shell Programming
	6.	C Programming: Introduction to C programming. Basic concepts.
	7.	C Programming: strings, arrays, functions
	8.	C Programming: structs and recursion
	9.	C Programming: pointers
	10.	C Programming: pointers and function pointers
	11.	C Programming: preprocessors and multifile programs
	12.	C Programming: core C libraries
	13.	C Programming: System-Level I/O
	14.	Makefiles
	15.	Final Exam/Project/Presentation Period
	16.	Final Exam/Project/Presentation Period
Required/Recommended Readings	Computer Systems: A Programmer's Perspective by Randal E. Bryant, David R. O'Hallaron (3rd Edition) Your UNIX/Linux: The Ultimate Guide by Sumitabha Das (3rd Edition) C How to Program by Paul Deitel and Harvey Deitel (8th Edition)	
Teaching Methods	Flipped classroom. Students work individually for lab practices and in groups for the project.	
Homework and Projects	Lab practices and Project	
Laboratory Work	Yes	

Computer Use	Required
Other Activities	-
Assessment Methods	Lab practices (20%), Project (20%), 2 Midterms (30% each)
Course Administration	Dr. Şeniz Demir **E-mail:** demirse@mef.edu.tr **Rules for attendance:** No attendance required. **Missing a lab practice:** Lab practices will start at the beginning of each lab session and end at the end of 2-hours period. Any submission not completed by the end of the session will receive partial credit. No grade will be given to a lab practice if the student does not attend that lab session. Provided that proper documents of excuse (e.g., health issues) are presented, each missed lab practice by the student will be given a grade which is equal to the average of all of the other lab practices (up to 2). No make-up will be given. A student who does not complete all lab practices will not receive any grade for 20% of the course (lab practices). **Missing the project:** No make-up for the project will be given. On late submissions, first day -10 points, second day -30 points, and third day -50 points deduction of the earned grade. No grade will be given to submissions after the third day. **Missing a midterm**: Provided that proper documents of excuse are presented, make-up for the missed midterm will be given. **A reminder of proper classroom behavior, code of student conduct**: YÖK Regulations **Statement on plagiarism**: YÖK Regulations

ECTS Student Workload Estimation

Activity	No/Weeks	Hours			Calculation	Explanation
	No/Weeks per Semester (A)	Preparing for the Activity (B)	Spent in the Activity Itself (C)	Completing the Activity Requirements (D)		
Lecture	14	2	3	0	70	A*(B+C+D)
Quizzes						A*(B+C+D)
Projects, Lab Practices	12	1	2	1	48	A*(B+C+D)
Midterm(s)	2	12	3	0	30	A*(B+C+D)
Final Examination						A*(B+C+D)
Total Workload					148	
Total Workload/25					5.92	
ECTS					**6**	