ABET 2000 Program Learning Outcomes

Engineering programs **must** demonstrate that their graduates have:

(a) an ability to apply knowledge of mathematics, science, and engineering
(b) an ability to design and conduct experiments, as well as to analyze and interpret data
(c) an ability to design a system, component, or process to meet desired needs
(d) an ability to function as an effective team member
(e) an ability to identify, formulate, and solve engineering problems
(f) an understanding of professional and ethical responsibility
(g) an ability to communicate effectively
(h) the broad education necessary to understand the impact of engineering solutions in a global and societal context
(i) a recognition of the need for, and an ability to engage in life-long learning
(j) knowledge of contemporary issues
(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Additional Computer Engineering Outcomes:

(l) Knowledge of Probability and Statistics and their applications in Computer Engineering
(m) Knowledge of Discrete Mathematics
(n) The ability to design and analyze IT (Information Technology) solutions for the Saudi Arabian market including the design and/or the integration of different hardware and software components

Course Learning Outcomes Guidelines

1. Course learning outcome should be **easily and directly measurable**.
2. Course learning outcomes suggested to be **4-6 outcomes**.
3. Each outcome should map to exactly one of the ABET program outcomes.
4. Include **Course Learning Outcomes Table** that includes the following for each outcome:
    - Outcome indicators and details: this describes the main course topics that will be focused on to achieve the outcome.
    - Suggested assessment methods and metrics.
    - A mapping between the course learning outcome and ABET program outcomes.
    - Each outcome will be given a rank as Low, High, Medium that correlates with the weight used for assessing the outcome. This will be used in the final mapping table between courses and ABET program outcomes.
    - When the course outcome weight is < 10%, it will be given a Low rank (L).
    - When the course outcome weight is between 10% and 20% it will be given a Medium rank (M).
    - When the course outcome weight is > 20% it will be given a High rank (H).

5. Include **Course Learning Outcomes Evaluation Table** that includes the following for each outcome:
    - **Outcome minimum weight**: this indicates the importance of the outcome in the course. It is the minimum weight from the total course score (out of 100) that must be used for assessing the outcome.
    - **Outcome weight**: this is to be filled by the instructor indicating how much weight was used by the instructor for assessing the outcome.
    - **Assessment Method**: this describes what methods are used to asses the outcome, the weight of each method, and the evidence of assessment.
    - **Class Average:** indicates the student's average performance in the outcome.
    - **Instructor Comments and Feedback:** this field is for the instructor to comment about the assessment of the outcome and suggest possible actions for improvement if necessary.

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
COLLEGE OF COMPUTER SCIENCES & ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

COE 205 Computer Organization & Assembly Language
Syllabus - Term 042

Catalog Description

Introduction to computer organization. Signed and unsigned number representation, character representation, ASCII codes. Assembly language programming, instruction format and types, memory and I/O instructions, dataflow, arithmetic, and flow control instructions, addressing modes, stack operations, and interrupts. Datapath and control unit design. RTL, microprogramming, and hardwired control. Practice of assembly language programming.
Prerequisite: COE 200 and ICS 201

Text Books & References:

• *Introduction to Assembly Language Programming: From 8086 to Pentium Processors*, Sivarama P. Dandamudi, et al., Springer Verlag, 1998. (ISBN: 0387985301).
• *Computer Systems Design and Architecture*, Vincent Heuring, Harry F. Jordan, Miles Murdocca, Addison Wesley 1997. (ISBN 0-8053-4330-X).
• *Assembly Language Programming and Organization of the IBM PC*, Ytha Yu and Charles Marut, McGraw Hill, 1992. (ISBN: 0-07-072692-2).
• *Online material: http://assembly*

Grading Policy

Laboratory	20%
Programming Assignments	10%
Quizzes	10%
Exam I	15%
Exam II	20%
Final	25%

Course Learning Outcomes

1. Knowledge of basic computer organization, information representation, and basic assembly language concepts.
2. Ability to analyze, design, implement, and test assembly language programs.
3. Ability to use tools and skills in analyzing and debugging assembly language programs.
4. Ability to design the datapath and control unit of a simple CPU.
5. Ability to demonstrate self-learning capability.
6. Ability to work in a team.

Course Topics

1. ***Introduction and Information Representation.*** **6 lectures**
   Introduction to computer organization. Instruction Set Architecture. Computer Components. Fetch-Execute cycle. Signed number representation ranges. Overflow.

2. ***Assembly Language Concepts.*** **6 lectures**
   Assembly language format. Directives vs. instructions. Constants and variables. I/O. INT 21H. Addressing modes.

3. ***8086 Assembly Language Programming.*** **20 lectures**
   Register set. Memory segmentation. MOV instructions. Arithmetic instructions and flags (ADD, ADC, SUB, SBB, INC, DEC, MUL, IMUL, DIV, IDIV). Compare, Jump and loop (CMP, JMP, Cond. jumps, LOOP). Logic, shift and rotate. Stack operations. Subprograms. Macros. I/O (IN, OUT). String instructions. Interrupts and interrupt processing, INT and IRET.

4. ***CPU Design.*** **12 lectures**
   Register transfer. Data-path design. 1-bus, 2-bus and 3-bus CPU organization. Fetch and execute phases of instruction processing. Performance consideration. Control steps. CPUMemory interface circuit. Hardwired control unit design. Microprogramming. Horizontal and Vertical microprogramming. Microprogrammed control unit design.

5. ***Instruction Set Formats.*** **1 lecture**
   Fixed vs. variable instruction format. Examples of instruction formats.

Course Learning Outcomes Table

Course Learning Outcomes	Outcome Indicators and Details	Assessment Methods and Metrics	ABET 2000 Criteria
1. Knowledge of basic computer organization, information representation, and basic assembly language concepts.	• Fetch-execute cycle • Data, address and control busses • Instruction Set Architecture • Number (unsigned and signed) and character representation • Addressing modes • Syntax, semantics, and effect on flags of Pentium instructions.	• Assignments • Quizzes • Exams	A(L)
2. Ability to analyze, design, implement, and test assembly language programs.	• Input/output. • Arithmetic and logic operations. • Flow-control structures. • Procedures. • Macros. • String manipulation. • Interrupt mechanism. • Implementation of Pseudo code algorithms in assembly language.	• Assignments • Quizzes • Exams	N(H)
3. Ability to use tools and skills in analyzing and debugging assembly language programs.	• Assembly language vs. machine language. • Assembling and linking assembly programs (including use of multiple files). • Use of debugger to analyze and debug programs. • Use of libraries.	• Lab work	K(M)

4. Ability to design the datapath and control unit of a simple CPU.	• Register transfer • Data path design: 1-bus, 2-bus and 3-bus CPU. • Derivation of control steps for assembly instructions. • Hardwired Control unit design • Microprogrammed control unit design. • Fixed vs. variable instruction format.	• Assignments • Quizzes • Exams	C(H)
5. Ability to demonstrate self-learning capability.	• Ability to learn a course topic alone (e.g. Macros) • Course Project may involve topics not studied in the course	• Assignments • Quizzes	I(L)
6. Ability to work in a team.	• Project is divided into separate parts that will be integrated for project completion.	• Lab project	D(L)

Course Learning Outcomes Evaluation Table

Outcome (minimum Weight)	Outcome Weight	Assessment Method								Class Average
O1	**26%**	**Assignments**	**Quizzes**	**Exam I**	**Exam II**	**Exam III**	**Final Exam**	**Lab Work**	**Project**	**71%**
(20%)			8%	10%	5%		3%			
		Evidence	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	
			Quiz#1-4, 6	Q1-3	Q1-2		Q1			
O2	**38%**	**Assignments**	**Quizzes**	**Exam I**	**Exam II**	**Exam III**	**Final Exam**	**Lab Work**	**Project**	**60%**
(30%)		10%		5%	15%				8%	
		Evidence	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	
				Q4-5	Q3-5					
O3	**10%**	**Assignments**	**Quizzes**	**Exam I**	**Exam II**	**Exam III**	**Final Exam**	**Lab Work**	**Project**	**80%**
(8%)								10%		
		Evidence	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	
								Exp# 1-13		

O4 (15%)	**22%**	**Assignments**	**Quizzes**	**Exam I**	**Exam II**	**Exam III**	**Final Exam**	**Lab Work**	**Project**	**75%**
							22%			
		Evidence	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	
							Q2-5			
O5 (2%)	**2%**	**Assignments**	**Quizzes**	**Exam I**	**Exam II**	**Exam III**	**Final Exam**	**Lab Work**	**Project**	**60%**
			2%							
		Evidence	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	
			Quiz#5							
O6 (2%)	**2%**	**Assignments**	**Quizzes**	**Exam I**	**Exam II**	**Exam III**	**Final Exam**	**Lab Work**	**Project**	**55%**
									2%	
		Evidence	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	**Evid.**	

Outcome Evaluation Example:

Class average for an outcome is computed by adding the average weights obtained from each assessment method used for the outcome divided by the total outcome weight. An example is shown below:

Class Average of Outcome 1 = [Quizzes (6) + Exam I (7) + Exam II (3.5) + Final Exam (2)]/26*100=18.5/26*100=71%

Instructor Comments and Feedback:

Instructors need to comment here about outcomes that they think were not achieved or there is a need for improvement. The instructor needs to suggest ways for improving outcome achievement in next course offerings.