

CS 1XA3: CS Experience & Practice: Basic concepts

Graduate Attributes and Indicators

George Karakostas

December 28, 2014

1 What the students should know and be able to do

1. Students should be able to
 - (a) Know the notions of variables and expressions
 - (b) Compute with numbers
 - (c) Manipulate sequences (Strings, Lists, and Files)
 - (d) Define functions and use call-by-value or call-by-reference
 - (e) Use decision structures and boolean expressions
 - (f) Use definite and indefinite loops
 - (g) Define classes and use objects in OOP
 - (h) Use recursion in algorithms
 - (i) Apply all basic concepts to build a more complex project

2 Mapping to Attributes with their Indicators

A01 Knowledge	
(3) Competence in Engineering Fundamentals	1a-li
(4) Competence in specialized engineering knowledge	1a-li
<hr/>	
A02 Analysis	
(5) Ability to identify the essential characteristics of a technical problem, including scope	1a-li
(6) Ability to identify reasonable assumptions (including identification of uncertainties and imprecise information) that could or should be made before a solution path is proposed	1a-li
(7) Ability to identify a range of suitable engineering fundamentals (including mathematical techniques) that would be potentially useful for analyzing a technical problem	1a-li
(8) Ability to decompose and organize a problem into manageable sub-problems	1g-li
(9) Ability to obtain substantiated conclusions as a results of a problem solution, including recognizing the limitations of the solutions	1a-li
(28) The ability to use of modern/state of the art tools	1a-li
<hr/>	
A03 Investigation	
(10) Able to recognize and discuss applicable theory knowledge base	1a-li
(11) Capable of selecting appropriate model and methods and identify assumptions and constraints	1a-li
<hr/>	
A04 Design	
(17) Recognizes and follows an engineering design process	1a-li
(18) Recognizes and follows engineering design principles	1a-li
(25) Properly documents and communicates processes and outcomes	1a-li
<hr/>	
A05 Tools	
<hr/>	
A06 Work	
<hr/>	
A07 Communication	
(37) Demonstrates an ability to respond to technical and non-technical instructions and questions	1a-li
(39) Demonstrates appropriate use of technical vocabulary	1a-li
<hr/>	
A08 Professionalism	
<hr/>	
A09 Impact	
<hr/>	
A10 Ethics	
<hr/>	
A11 Economics	
<hr/>	
A12 Learning	
<hr/>	
A13 Sustainability	

3 Course work

The course work consists of 11 weekly labs (100%).

4 Prerequisites

No prerequisites.

5 Learning outcomes

Topic	Below	Marginal	Meets	Exceeds
1a	doesn't know what variables and expressions are	has an incomplete grasp of variables and expressions	knows what variables and expressions are and how they are used	can identify complicated expressions
1b	doesn't know how to use numbers in computation	can use some numerical expressions but has difficulties with different numerical types	can use the correct numerical operations for different numerical types	turns numerical data into their correct type before applying numerical operations
1c	doesn't know how to manipulate sequences	knows how to manipulate some kinds but not all kinds of sequences	knows how to manipulate all kinds of sequences	can perform sophisticated manipulations of sequences
1d	cannot properly define functions	can define functions but cannot use properly call-by-value or call-by-reference	can define functions and use call-by-value or call-by-reference	can do structured and sophisticated function definitions and uses of call-by-value or call-by-reference

1e	cannot use decision structures and boolean expressions	can use some decision structures and boolean expressions	can use decision structures and boolean expressions	can make a sophisticated use of decision structures and boolean expressions
1f	cannot use definite and indefinite loops	can use some definite and indefinite loops	can use correctly definite and indefinite loops	can make a sophisticated use of definite and indefinite loops
1g	cannot properly define classes and use objects in OOP	can define classes and cannot use properly objects in OOP	can define classes and use objects in OOP	can do structured and sophisticated class definitions and uses of objects in OOP
1h	cannot use recursion in algorithms at all	can use some simple recursive definitions	can use recursion in algorithms	can make a sophisticated use of recursion in algorithms
1i	cannot apply basic concepts to build a more complex project	can apply basic concepts to partially build a more complex project	can apply basic concepts to build a complex project	can make a sophisticated application of all basic concepts to build a complex project