ABET Criterion 3: Student Outcomes

The faculty of Computer Science has adopted Criteria (a)-(i) and additional Criteria (j)-(k) as the Student Outcomes for our program.

GENERAL CRITERION 3. STUDENT OUTCOMES

`http://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2015-2016/#outcomes`

```
The program must have documented student outcomes that prepare
graduates to attain the program educational objectives. There must be a
documented and effective process for the periodic review and revision of these
student outcomes.  The program must enable students to attain, by the time of
graduation:

(a) An ability to apply knowledge of computing and mathematics appropriate to
the programs student outcomes and to the discipline

(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

(d) An ability to function effectively on teams to accomplish a common goal

(e) An understanding of professional, ethical, legal, security and social issues and responsibilities

(f) An ability to communicate effectively with a range of audiences

(g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

(h) Recognition of the need for and an ability to engage in continuing professional development

(i) An ability to use current techniques, skills, and tools necessary for computing practice.

These program criteria apply to computing programs using computer science or
similar terms in their titles.

The program must enable students to attain, by the time of graduation:

(j) An ability to apply mathematical foundations, algorithmic principles, and
computer science theory in the modeling and design of computer-based systems in
a way that demonstrates comprehension of the tradeoffs involved in design
choices. [CS]

(k) An ability to apply design and development principles in the construction
of software systems of varying complexity. [CS]
```

Assessment Guidelines for our Computer Science (University of Kentucky):

- Evaluation of a student learning outcome (SLO) for each of the selected courses requires 3 artifacts, if possible. These could be three separate questions on an exam, three different assignments, or 3 separate components of a single assignment. The preference is on different types of assignments and on assignments that are in a controlled environment: exams, labs, etc. (homework assignments are less desirable). The evaluated assignments/tasks for students should be prepared so there is are sufficient volume and granularity allowing the evaluator for meaningful application of the rubrics.
- Complete one rubric for each selected student; not all students have to be assessed. Pick a random population. Based on a discussion, we decided that we should assess 20% of the students but no less than 10. In practice, when the class is smaller than 50 students, select a sample of $\min(sizeClass, 10)$ students, otherwise select one-fifth.
- The random sample must be selected BEFORE grading the assignment: evaluation for ABET is based on the rubrics and is separate from grading. Make sure that the population is random.
- If an artifact is blank, you can throw it out but replace with another random sample.
- Keep documentation of the assessment and prepare summary of the results:
  - Description of the assignment
  - Selected samples and their evaluation with rubrics
  - A report: results, how many students attained *exceed expectations* and *meet expectations* levels, discuss the results.

			A	B	C	D	E	F	G	H	I	J	K	
CS 100	**Comp Sci Profession**	1	–	–	+	–	–	–	+	+	–	–	–	<u>CS 100</u>
CS 115	**Introduction to Computer Programming**	3	+	–	$\oplus$	–	+	–	–	–	–	–	–	CS 115*
CS 215	**Intro to Program Design Abstraction**	4	+	$\oplus$	+	–	–	–	–	–	–	–	–	CS 215
CS 216	**Introduction to Software Engineering**	3	+	+	+	–	–	–	–	–	–	–	–	CS 216*
CS 270	Systems Programing	4	–	–	+	–	–	–	–	–	$\oplus$	–	$\oplus$	CS 270
CS 275	**Discrete Mathematics**	4	$\oplus$	–	–	–	–	–	–	–	–	+	–	CS 275*
MA 113	Calculus I	4	–	–	–	–	–	–	–	–	–	–	–	MA 113
MA 114	Calculus II	4	–	–	–	–	–	–	–	–	–	–	–	MA 114
PHY 231	General University Physics	4	–	–	–	–	–	–	–	–	–	–	–	PHY 231
PHY 241	General University Physics Laboratory	1	–	–	–	–	–	–	–	–	–	–	–	PHY 241
PHY 232	General University Physics	4	–	–	–	–	–	–	–	–	–	–	–	PHY 232
PHY 242	General University Physics Laboratory	1	–	–	–	–	–	–	–	–	–	–	–	PHY 242
MA 213	Calculus III	4	–	–	–	–	–	–	–	–	–	–	–	MA 213
EE 280	**Design of Logic Circuits**	3	–	–	+	–	–	–	–	–	+	+	–	EE 280
STA 281	Prob and Stat Using Interactive Comp Tech	3	–	–	–	–	–	–	–	–	–	–	–	STA 281
CS 315	**Algorithm Design and Analysis**	3	+	$\oplus$	+	–	–	–	–	–	–	$\oplus$	–	CS 315*
CS 316	Web Programming	3	+	+	+	+	–	–	–	–	$\oplus$	–	$\oplus$	CS 316
CS 321	**Introduction to Numerical Methods**	3	+	$\oplus$	–	–	–	–	–	–	–	+	–	CS 321*
MA 322	Matrix Algebra and its Applications	3	–	–	–	–	–	–	–	–	–	–	–	MA 322
CS 335	Graphics and Multimedia	3	+	–	+	$\oplus$	–	–	–	–	+	+	+	CS 335
CS 368	Cryptology	3	$\oplus$	–	–	–	–	+	+	–	–	$\oplus$	–	CS 368
CS 371	Intro to Networking	3	–	–	$\oplus$	–	–	–	–	–	$\oplus$	–	+	CS 371*
CS 375	**Logic and Theory of Computing**	3	$\oplus$	–	–	–	–	–	–	$\oplus$	–	+	–	CS 375*
CS 368	Cryptology	3	$\oplus$	–	–	–	–	+	+	–	–	$\oplus$	–	CS 368*
CS 380	**Microcomputer Organization**	3	–	+	+	–	–	–	–	–	–	+	–	CS 380
CS 395	Independent Studies	2	–	–	–	–	–	–	–	–	–	–	–	CS 395
CS 405G	Introduction to Database Systems	3	+	$\oplus$	+	–	+	–	–	–	–	–	–	CS 405G*
CS 441G	Compilers for Algorithmic Languages	3	+	+	+	–	–	–	–	–	+	+	–	CS 441G
CS 450G	Fundamentals of Programming Languages	3	–	+	–	–	+	–	–	–	–	+	–	CS 450G
CS 463G	Logic and Artificial Intelligence	3	+	+	–	–	$\oplus$	+	$\oplus$	+	+	+	–	CS 463G
CS 460G	Machine Learning	3	+	+	+	–	–	+	–	–	+	+	–	CS 460G
CS 470G	Intro Operating Systems	3	–	–	$\oplus$	–	–	–	–	–	+	$\oplus$	–	CS 470*
CS 471G	Networking and Dist Op Sys	3	–	–	+	–	–	–	–	–	+	–	+	CS 471G
CS 485G	Topics in CS	3	–	–	–	–	–	–	–	–	–	–	–	CS 485G
CS 485G	subtitle: Systems Programming	3	–	–	$\oplus$	–	–	–	–	–	+	$\oplus$	–	CS 485G*
CS 485G	subtitle: Concurrency	3	–	–	$\oplus$	–	–	–	–	–	+	$\oplus$	–	CS 485G*
CS 485G	subtitle: Data Mining	3	+	+	–	–	–	–	+	–	+	+	+	CS 485G*
CS 485G	subtitle: iOS Programming	3	–	–	$\oplus$	–	–	–	–	–	+	$\oplus$	–	CS 485G*
CS 498	Soft Engr Senior Design	3	–	–	–	+	$\oplus$	–	–	–	+	–	–	CS 498*
CS 499	**Senior Design Project**	3	+	+	+	$\oplus$	+	$\oplus$	+	+	+	+	+	CS 499*
CS 505	Intermediate Topics in DB	3	+	–	+	–	–	–	–	–	–	+	+	CS 505*
CS 515	Algorithms	3	+	+	+	–	+	–	–	–	–	+	–	CS 515
CS 535	Intro to Comp Graphics	3	+	–	+	+	$\oplus$	–	–	–	+	+	+	CS 535
CS 537	Numerical Analysis	3	+	$\oplus$	–	–	–	–	–	–	–	+	–	CS 537
CS 541	Compiler Design	3	+	+	+	+	–	–	–	–	+	+	+	CS 541
CS 570	Operating Systems	3	–	–	+	–	–	–	–	–	+	+	–	CS 570
CS 571	Computer Networks	3	–	–	+	–	–	–	–	–	+	+	–	CS 571
CS 575	Models of Computation	3	+	+	–	–	–	–	–	–	–	+	–	CS 575*
CS 585	Intermediate Topics in CS	3	–	–	–	–	–	–	–	–	–	–	–	CS 585
CS 585	subtitle: Ethics	3	–	–	–	–	–	–	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	CS 585

Table 1: Courses-to-SOL mapping for the CS UK curriculum to assess Student Outcomes (v. 2016-01-11): + the course is used in the assessment of this SLO; $\oplus$ the course is used in the assessment of this SLO in Spring 2016; – the course is not used in the assessment of this SLO.