QFO-AP-FI-MO02	اسم النموذج: خطة تدريس مادة دراسية Course Syllabus	جامعة فيلادلفيا Philadelphia University
رقم الاصدار :1 Revision 1	الجهة المصدرة: كلية تكنولوجيا المعلومات	
التاريخ :2017/11/05	الجهة المدققة: عمادة التطوير والجودة	
عدد صفحات النموذج: 3		

2018-2019-1

Course title: Formal Methods	**Course code:** 0750784
Course level: MSc	**Course prerequisite:** Background in programming and in software engineering
Lecture time:	**Credit hours:** 3

Academic Staff Specifics				
Name	**Rank**	**Office location**	**Office hours**	**e-mail address**

Course description: Formal methods in software engineering address various topics. Among these topics, the following are pointed out in the CS2013 Final Report endorsed by ACM and the IEEE Computer Society.

- Role of formal specification and analysis techniques in the software development cycle.
- Program assertion languages and analysis approaches.
- Formal approaches to software modeling and analysis.
- Model checkers.
- Model finders.
- Tools in support of formal methods.

In this MSc course you will be seamlessly introduced to the practical applications of formal methods within the software development process with the broad aim of understanding how the use of such methods even if pragmatically applied might conduct to the construction of maintainable and high quality software systems while keeping acceptable development costs.

Teaching methods: Lectures, laboratories, seminars, workshops

Learning outcomes:

At the end of the module, you should be able to:

A. Knowledge and understanding

A1. Understand the benefits of formal methods for software development
A2. Distinguish property-oriented an model-oriented formal methods
A3. Write formal specifications

B. Intellectual skills

B1. Prove properties about formal specifications
B2. Use appropriate techniques and tools in (formal) software development

C. Practical skills

C1. Develop small programs out of specifications
C2. Apply appropriate approaches to various categories of software engineering problems

D. Transferable skills and personal qualities

D1. Prepare structured technical reports for lab work assignment.
D2. Use the Internet to research articles on the subjects of this course; Prepare a state-of-the art reports; Deliver verbal communications.

Assessment of learning outcomes:

Learning outcomes A1, A2, B1, and B2 are assessed by examinations; Learning outcomes A3, C1, and C2 are assessed by lab work; Learning outcomes D1 and D2 are assessed by seminars and/or workshops.

Assessment instruments:

Allocation of Marks	
Assessment Instruments	**Marks**
Midterm examination	**30%**
Final Exam (written unseen exam)	**40 %**
Assignments (lab work), research work	**10% + 20%**
Total	**100%**

** Make-up exams will be offered for valid reasons only with consent of the Dean. Make-up exams may be different from regular exams in content and format.*

Documentation and academic honesty:

- Handed reports must be presented according to the style specified in the assignment guide.
- Protection by copyright.
- Avoiding plagiarism: any stated plagiarism leads to an academic penalty.

Course academic calendar

Week	Basic and support material to be covered	Assignments/ Research work
(1)	Introduction to the use of formal methods in Software Engineering	
(2)	Algebraic Specifications Specification of stacks, queues and lists	
(3)	Algebraic Specifications (cont'd) Specification of sets, strings and editors	
(4)	Algebraic Specifications (cont'd) Foundations: signature, algebra, terms, substitution, properties of abstract data types	**Assignment 1:** Research work
(5)	Algebraic Specifications (cont'd) Equation-based reasoning, proofs by induction	
(6)	Algebraic Specifications (cont'd) Modules, genericity, parameterized specifications	
(7)	Algebraic Specifications (cont'd) Abstract implementations	**Assignment 2:** lab work
(8)	**Midterm Exam**	
(9)	The Z specification language Examples	
(10)	The Z specification language (cont'd) Elements of Z specifications, type concept, notations, generic definitions, relations, functions	
(11)	The Z specification language (cont'd) Schemata, proving of specifications properties	
(12)	Z and Object-Z (OZ)	**Assignment 3:** lab work
(13)	Introduction to modelling with Petri nets	
(14)	CSP	
(15)	CSP/OZ	
(16)	**Final Exam**	

Expected workload: On average you need to spend 6 hours of study and preparation for each lecture.

Attendance policy: Absence from lectures and/or tutorials shall not exceed 15%. Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and shall receive a mark of zero for the course. If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

Literature and supporting material:

1. C. A. R. Hoare: Communicating Sequential Processes, 2015
2. T. Mossakovski et al.: The Heterogeneous Tool Set, University of Bremen, 2013
3. D. Sannella & A. Tarlecki: Foundations of Algebraic Specifications and Formal Software Development, Springer, 2012
4. V.S. Alagar, K. Periyasamy: Specification of Software Systems, Graduate Texts in Computer Science, Springer-Verlag, 2nd Edition, 2011.
5. Sommerville: Software Engineering, 9th Edition, Pearson Addison Wesley, 2010.
6. G. Smith: The Object-Z specification language, Kluwer Academic Publisher, 2000.
7. M. Hinchey & J. Brown: Industrial-Strength Formal Methods in Practice, Springer, 1999.
8. G. Rozenberg: Graph Grammars and computing by graph transformations, Volume 1: Foundations, World Scientific, 1997.
9. J. Woodcock & J. Davies, Using Z, Specification, Refinement, and Proof, Prentice Hall International Series for Computer Science, 1996.
10. W. Reisig: A Primer in Petri Net Design, Springer, 1992.
11. H. Ehrig & B. Mahr: Fundamentals of algebraic specifications I, Springer, 1985.

Supporting tools and environments:

Maude (algebraic specifications)
CafeOBJ (algebraic specification and verification)
TOZE (Object-Z)

Reading Material:

Selected research papers

Electronic material:

http://ifipwg13.informatik.uni-bremen.de
http://www.philadelphia.edu.jo/academics/bettaz/