QFO–AP–FI–MO02	Course Syllabus :اسم النموذج	جامعة فيلادلفيا
رقم الاصدار : 1 (Revision)	الجهة المصدرة: كلية تكنولوجيا المعلومات	
التاريخ :2017/11/05	الجهة المدققة: عمادة التطوير والجودة	Philadelphia University
عدد صفحات النموذج:		

Faculty of Information Technology
Department of Software Engineering
2020-2021.2

Course Syllabus	
Course Title: Formal Methods	**Course code: 0721717**
Course Level: Master	**Course prerequisite (s) and/or corequisite (s):**
Lecture Time: Sunday, 16 – 19	**Credit hours: 3**

Academic Staff Specifics				
Name	**Rank**	**Office Location**	**Office Hours**	**E-mail Address**
Dr. Said Ghoul	Professor	IT-307	Sunday, 15:00-16:00	sghoul@philadelphia.edu.jo

Course/ module description

The module aims at introducing the students to the application of formal methods to the practice of software engineering. Formal Methods refer to a variety of mathematical modeling techniques, which are used both to model the behavior of a computer system and to verify that the system satisfies design, safety and functional properties. This is a course in formal mechanisms for specifying, validating and verifying, and constructing correct software systems.

Course/ module objectives

This course aims to:

- Introduce Formal construction methods fundamentals
- Study in depth Algebraic Specification
- Study in depth Z Specification
- Study in depth Graph Rewriting Systems
- Study in general SDL
- Study of others methods: UML OZ, Petri Nets, …

Course/ module components

- **Books/ papers (title , author (s), publisher, year of publication)**

1. Jiacun Wang Formal Methods in Computer Science Chapman and Hall/CRC, 2019
2. Gerard O'Regan Concise Guide to Formal Methods: Theory, Fundamentals and Industry Applications, Springer, 2017

- **Support material (s) (vcs, acs, etc)**
  Courses slides are available (course of Dr. Bernhard Westfechtel with authorization)
- **Study guide (s) (if applicable)**
  The courses is composed by to parts:
  - Part 1/ Algebraic language
  - Part 2/ Z language
  - Part 3/ Graph transformations
  - Part 4/ SDL
  - Part 5/Others ethods
- **Homework and laboratory guide (s) if (applicable)**
  - attached Homework and Practical work sheet

Teaching methods

Duration: 16 weeks, 45 hours in total. Lectures: 15 hours, 1 per week. Tutorial: 30 hours. Laboratories, HW: 15 hours in total, 1-hour per week (personal). The last week is reserved to practical works examination.

Learning outcomes

A student completing this module unit should be able to:

- A. Knowledge and understanding
  1. Understand A wide range of principles and tools available to the software engineer and system developer, such as Formal methods in Software Engineering. All these directions informed by research.
  2. Understand the professional and ethical responsibilities of the practicing computer professional including understanding the need for quality.
  3. Understand the application of computing in a business context

- B. Cognitive skills (thinking and analysis).
  1. Solve a wide range of problems related to the Formal methods in Software engineering
  2. Analysis and Design of Formal specification of small size software.
  3. Be able to design, write and debug Formal specifications in appropriate languages.

- C. Communication skills (personal and academic).
  1. Plan and undertake a major individual project, and prepare and deliver coherent and structured verbal and written technical report.
  2. Be able to display an integrated approach to the deployment of communication skills, use IT skills and display mature computer literacy; strike the balance between self-reliance and seeking help when necessary in new situations, and display personal responsibility by working to multiple deadlines in complex activities.

- D. Practical and subject specific skills (Transferable Skills).
  1. Be able to deal effectively with new Formal methods in Software engineering, techniques, and tools

Learning Outcomes A**chievement**

Developed: Lecture:A1..A3.B1..B3
Tutorial, HW: B3, C1, C2, D1
Assessed: by Exam: A3, B3, by Homework: all others ILOs

Assessment instruments

Allocation of Marks	
Assessment Instruments	**Mark**
Midterm examination	**30%**
Final Exam (written unseen exam)	**50 %**
Reports, research projects	**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.*

Practical Submissions

The assignments that have work to be assessed will be given to the students in separate documents including the due date and appropriate reading material.

Documentation and Academic Honesty

Submit your homework covered with a sheet containing your name, number, course title and number, and type and number of the home work (e.g. assignment, and project).

Any completed homework must be handed in the class on the due date. After the deadline "zero" will be awarded. You must keep a duplicate copy of your work because it may be needed while the original is being marked.

You should hand in with your assignments:

2. A brief report to explain your findings.
3. Your solution of given problem

For the research report, you are required to write a report similar to a scientific research paper. It should include:

4. *Abstract:* It describes the main synopsis of your paper.
5. *Introduction:* It provides background information necessary to understand the research and getting readers interested in your subject. The introduction is where you put your problem definition, summary of contribution, related work, and is likely where the bulk of your sources will appear.
6. *Methods (Algorithms and Implementation):* Describe your methods here. Summarize the algorithms (if any) generally, highlight features relevant to your project, and refer readers to your references for further details. Information from sources must be rephrased in own words, "copy-and-paste" from documents, found for example on the Internet, is NOT allowed. It is allowed to use short quotations, or figures, from other documents, but then the source MUST be clearly stated in the reference list (please check copy rights). Papers not fulfilling these rules will be failed.

7. *Results and Discussion (Benchmarking and Analysis):* This section is the most important part of your paper. It is here that you demonstrate the work you have accomplished on this project and explain its significance. The quality of your analysis will impact your final grade more than any other component on the paper. You should therefore plan to spend the bulk of your project time not just gathering data, but determining what it ultimately means and deciding how best to showcase these findings.
8. *Conclusion:* The conclusion should give your reader the points to "take home" from your paper. It should state clearly what your results demonstrate about the problem you were tackling in the paper. It should also generalize your findings, putting them into a useful context that can be built upon. All generalizations should be supported by your data, however; the discussion should prove these points, so that when the reader gets to the conclusion, the statements are logical and seem self-evident.
9. *Bibliography:* Refer to any reference that you used in your assignment. Citations in the body of the paper should refer to a bibliography at the end of the paper.

- **Protection by Copyright**

1. Coursework, laboratory exercises, reports, and essays submitted for assessment must be your own work, unless in the case of group projects a joint effort is expected and is indicated as such.
2. Use of quotations or data from the work of others is entirely acceptable, and is often very valuable provided that the source of the quotation or data is given. Failure to provide a source or put quotation marks around material that is taken from elsewhere gives the appearance that the comments are ostensibly your own. When quoting word-for-word from the work of another person quotation marks or indenting (setting the quotation in from the margin) must be used and the source of the quoted material must be acknowledged.
3. Sources of quotations used should be listed in full in a bibliography at the end of your piece of work.

- **Avoiding Plagiarism**

1. Unacknowledged direct copying from the work of another person, or the close paraphrasing of somebody else's work, is called plagiarism and is a serious offence, equated with cheating in examinations. This applies to copying both from other students' work and from published sources such as books, reports or journal articles.
2. Paraphrasing, when the original statement is still identifiable and has no acknowledgement, is plagiarism. A close paraphrase of another person's work must have an acknowledgement to the source. It is not acceptable for you to put together unacknowledged passages from the same or from different sources linking these together with a few words or sentences of your own and changing a few words from the original text: this is regarded as over-dependence on other sources, which is a form of plagiarism.
3. Direct quotations from an earlier piece of your own work, if not attributed, suggest that your work is original, when in fact it is not. The direct copying of one's own writings qualifies as plagiarism if the fact that the work has been or is to be presented elsewhere is not acknowledged.

4. Plagiarism is a serious offence and will always result in imposition of a penalty. In deciding upon the penalty the Department will take into account factors such as the year of study, the extent and proportion of the work that has been plagiarized, and the apparent intent of the student. The penalties that can be imposed range from a minimum of a zero mark for the work (without allowing resubmission) through caution to disciplinary measures (such as suspension or expulsion).

Course/Module Academic Calendar

week	Basic and support material to be covered	Homework
(1)	Chap. 1 /Introduction to formal methods Formal methods in Software Engineering, Basic Notations and classification	
(2)	Chap.2 /Algebraic methods Algebraic method (1) : Foundations	
(3, 4)	Algebraic method (2) : Modules	
(5, 6)	Algebraic method (3) : Concrete specifications	
(7)	Algebraic method (4) : Abstract Implementations	**Assign. 1**
(8)	**Mid Exam**	
(9)	Chap.3 / Z methods Z method (1): Foundations, Notations, Schemata	
(10)	Z method (2): Schemata, Sample (tutorial)	
(11)	Z method (3): Refinement, Object Z	
(12)	Z method (3): Tutorials	
(13)	Chap. 4 / Specification and Description Language - SDL Foundation, Notations, Examples, Extended concepts	**Assign.2**
(14)	Chap. 5 / Graph Rewriting Systems Introductory example, Theoretical foundations, Specification with PROGRES, Attributed graph and graph schemata, Graph rewrite rules, Control structures,	
(15)	Chap. 6 / Others Methods UML OZ, Petri Nets	
(16)	Practical Exam	

Expected workload

On average students need to spend 2 hours of study and preparation for each 50-minute lecture/tutorial.

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.

Module references

Books

- John Fitzgerald, Ian Hayes, Andrzej Tarlecki. FM 2005: formal methods : International Symposium of Formal Methods Europe Proceedings, LNCS 3582, Springer, 2005
- Jean François Monin, Michael Gerard Hinchey. Understanding formal methods. Springer Verlag, 2003
- Jim Woodcock, Jim Davis. *Using Z: Specification, Refinement and Proof*. Prentice-Hall International 1996
- V Heitmeyer and Mandrioli (eds), Formal methods for real-time computing, John Wiley, 1996

Research Papers

1. Formal Methods Application: An Empirical Tale of Software Development (Sobel) 2002
2. Correctness by Construction: Developing a Commercial Secure System (Hall and Chapman), IEEE Software Jan/Feb 2002
3. Cost Effective Use of Formal Methods in Verification and Validation (Kuhn, Chandramouli and Butler) Foundations 2002 V&V Workshop
4. Industrial Strength Exception Freedom (Amey and Chapman) 2002
5. Formal Methods Diffusion: Prospects (Adelard) 2000
6. Automated Deduction Looking Ahead (Loveland), AI Magazine, Spring 1999

7. Formally Verifying IEEE Compliance of Floating-Point Hardware (Leary, Zhao, Gerth, and Seger)Intel Technology Journal 1999
8. From Formal Models to Formally Based Methods: An Industrial Experience (Ciapessoni), ACM Transactions on Software Engineering and Methodology, Jan. 1999.
9. A mechanically checked proof of correctness of the AMD-K5* floating point square root microcode (Russinoff), Formal Methods in System Design 1999
10. The Industrial Use of Formal Methods: Was Darwin Right? (Miller) Second IEEE Workshop on Industrial Strength Formal Specification Techniques Oct 1998
11. Formal Verification In a Commercial Setting (Kurshan) Bell Laboratories DAC 97
12. Formal Methods: State of the Art and Future Directions (Clarke and Wing) ACM Computing Surveys 1996

Websites

http://ecourse.philadelphia.edu.jo/login/index.php → prof. S. Ghoul → Formal Methods
http://www.se.rwth-aachen.de/tikiwiki/tiki-index.php%3Fpage_ref_id=333.html → Dr. Bernhard Westfechtel (shared material)
http://www.springer.com/engineering/circuits+&+systems/journal/10703
http://www.brics.dk/formal-methods/

- Virtual library on formal methods