هيئة تقويم التعليم والتدريب
Education & Training Evaluation Commission
اعتماد
NCAAA
2024
TP-153

# Course Specification

## (Bachelor)

| | |
|---|---|
| **Course Title:** | Formal Methods |
| **Course Code:** | SE4713 |
| **Program:** | BSc in Software Engineering |
| **Department:** | Software Engineering |
| **College:** | College of Computing |
| **Institution:** | Umm Al Qura University |
| **Version:** | 1.0 |
| **Last Revision Date:** | 22/04/2025 |

# Table of Contents

## A. General information about the course:

### 1. Course Identification

| 1. Credit hours: (3) | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| **2. Course type** | | | | | |
| **A.** | ☐University | ☐College | ☒ Department | ☐Track | ☐Others |
| **B.** | ☐ Required | | ☒ Elective | | |
| **3. Level/year at which this course is offered: ( 3rd year/ 5th or 6th level) or ( 4th year/8th level)** | | | | | |
| **4. Course General Description:** | | | | | |
| This course is dedicated to studying formal techniques for building reliable software. The course will first teach the mathematical foundations of formally representing programs as mathematical objects, and how to reduce program verification to a mathematical theorem. Using this foundation, we will introduce various abstractions, such as contract based programming, as a way of formally specifying properties of code that facilitates modular and reliable program development.<br><br>We will cover a range of program verification and bug-finding techniqes for sequential and more challenging programs (e.g., concurrent, non-deterministic, or probabilistic). The course will be a mixture of theory and practice: the students will study practical applications using tools that prove important properties, such as safety or termination, using abstraction based techniques, model-checking, and developing programs using contracts. | | | | | |
| **5. Pre-requirements for this course** (if any)**:** | | | | | |
| **CS2105 - Data Structures and Algorithms with Java** | | | | | |
| **6. Co-requisites for this course** (if any)**:** | | | | | |
| N/A | | | | | |
| **7. Course Main Objective(s):** | | | | | |
| By the end of the course, students will:<br>1. Understand the theoretical foundations of formal methods and their role in software verification.<br>2. Apply logical reasoning to specify and prove the correctness of software programs.<br>3. Use symbolic execution, dataflow analysis, and abstract interpretation to analyze and optimize program behavior.<br>4. Verify system properties, including concurrency, using model-checking techniques. | | | | | |

5. Gain practical experience with formal tools and methodologies to ensure software safety and reliability.

## 2. Teaching mode (mark all that apply)

| No | Mode of Instruction | Contact Hours | Percentage |
|---|---|---|---|
| 1 | Traditional classroom | 45 | 100% |
| 2 | E-learning | 0 | 0 |
| 3 | Hybrid<br>• Traditional classroom<br>• E-learning | 0 | 0 |
| 4 | Distance learning | 0 | 0 |

## 3. Contact Hours (based on the academic semester)

| No | Activity | Contact Hours |
|---|---|---|
| 1. | **Lectures** | 45 |
| 2. | **Laboratory/Studio** | 0 |
| 3. | **Field** | 0 |
| 4. | **Tutorial** | 0 |
| 5. | **Others (specify)** | 0 |
| **Total** | | 45 |

## B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

| Code | Course Learning Outcomes | Code of PLOs aligned with the program | Teaching Strategies | Assessment Methods |
|---|---|---|---|---|
| **1.0** | **Knowledge and understanding** | | | |
| 1.1 | Understand the principles of propositional logic and first-order logic in the context of program analysis. | K1 | Lecture, Exercise | Quiz, Exams, Assignments |
| 1.2 | Learn the syntax and semantics of simple imperative programs, | K2 | Lecture, Exercise | Quiz, Exams, Assignments |

| Code | Course Learning Outcomes | Code of PLOs aligned with the program | Teaching Strategies | Assessment Methods |
|---|---|---|---|---|
| | including symbolic execution techniques. | | | |
| 1.3 | Explore model checking methods for analyzing program properties, including handling concurrency. | K3 | Lecture, Exercise | Quiz, Exams, Assignments |
| **2.0** | **Skills** | | | |
| 2.1 | Apply dataflow analysis and abstract interpretation to evaluate program behavior. | S3 | Lecture, Exercise | Quiz, Exams, Assignments |
| 2.2 | Use Hoare logic to reason about program correctness and termination. | S1 | Lecture, Exercise | Quiz, Exams, Assignments |
| 2.3 | Analyze and reason about pointers and their implications in program semantics. | S3 | Lecture, Exercise | Quiz, Exams, Assignments |
| **3.0** | **Values, autonomy, and responsibility** | | | |
| 3.1 | Appreciate the importance of formal methods in ensuring program correctness and reliability. | V1 | Lecture, Exercise | Quiz, Exams, Assignments |
| 3.2 | Recognize the value of systematic approaches like model checking for verifying complex systems. | V1 | Lecture, Exercise | Quiz, Exams, Assignments |
| 3.3 | Embrace logical reasoning and abstraction as tools to improve software quality and safety. | V1 | Lecture, Exercise | Quiz, Exams, Assignments |

## C. Course Content

| No | List of Topics | Contact Hours |
|---|---|---|
| **1.** | Introduction | **3** |
| **2.** | Background and Propositional Logic | **3** |

| | | |
|---|---|---|
| 3. | Simple Imperative Programs, Syntax and Semantics | 6 |
| 4. | Symbolic Execution | 3 |
| 5. | Dataflow analysis and abstract interpretation | 6 |
| 6. | First Order Logic | 6 |
| 7. | Hoare Logic | 3 |
| 8. | Reasoning About Pointers | 6 |
| 9. | Reasoning About Termination | 3 |
| 10. | Model Checking | 3 |
| 11. | Model Checking: Concurrency | 3 |
| Total | | 45 |

## D. Students Assessment Activities

| No | Assessment Activities * | Assessment timing (in week no) | Percentage of Total Assessment Score |
|---|---|---|---|
| 1. | Quizzes | 2-14 | 15 |
| 2. | Projects | 2-14 | 15 |
| 3. | Assignments | 2-14 | 10 |
| 4. | Mid Term | 7 | 20 |
| 5. | Final Exam | 16-17 | 40 |

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

## E. Learning Resources and Facilities

### 1. References and Learning Resources

| | |
|---|---|
| Essential References | • Apt, K. R., de Boer, F. S., & Olderog, E.-R. (2009). *Verification of sequential and concurrent programs* (3rd ed.). Springer. ISBN 978-1848827448. |
| Supportive References | • Apt, K. R., & de Boer, F. S. (2009). *Verification of sequential and concurrent programs* (3rd ed.). Springer. ISBN 978-1848827455.<br>• Bradley, A. R., & Manna, Z. (2007). *The calculus of computation: Decision procedures with applications to verification*. Springer. ISBN 978-3540741121.<br>• Nielson, F., Nielson, H. R., & Hankin, C. (1999). *Principles of program analysis*. Springer. ISBN 978-3540654100.<br>• Møller, A., & Schwartzbach, M. I. (2018). *Static program analysis*. Springer. ISBN 978-3319907895. |

| | |
|---|---|
| | • Vreeswijk, G. A. W. (Ed.). (2008). *Automated reasoning*. Springer. ISBN 978-1402062439.<br>• Baier, C., & Katoen, J.-P. (2008). *Principles of model checking*. MIT Press. ISBN 978-0262026499.<br>• Rival, X., & Yi, K. (2020). *Introduction to static analysis: An abstract interpretation perspective*. MIT Press. ISBN 978-0262043847. |
| **Electronic Materials** | |
| **Other Learning Materials** | |

## 2. Required Facilities and equipment

| Items | Resources |
|---|---|
| **facilities**<br>(Classrooms, laboratories, exhibition rooms, simulation rooms, etc.) | Traditional Classroom |
| **Technology equipment**<br>(projector, smart board, software) | Multimedia Projector |
| **Other equipment**<br>(depending on the nature of the specialty) | N/A |

## F. Assessment of Course Quality

| Assessment Areas/Issues | Assessor | Assessment Methods |
|---|---|---|
| Effectiveness of teaching | Students | Direct, Indirect |
| Effectiveness of Students assessment | Faculty, Peer reviewer | Direct, Indirect |
| Quality of learning resources | Faculty, Course coordinator | Direct, Indirect |
| The extent to which CLOs have been achieved | Course coordinator, Program management committee | Direct |
| Other | | |

**Assessors** (Students, Faculty, Program Leaders, Peer Reviewers, Others (specify)

**Assessment Methods** (Direct, Indirect)

## G. Specification Approval

| | |
|---|---|
| **COUNCIL /COMMITTEE** | **SOFTWARE ENGINEERING DEPARTMENT COUNCIL** |
| **REFERENCE NO.** | **THE 17TH MEETING FOR THE ACADEMIC YEAR 1446H** |
| **DATE** | **22/04/2025** |