TO: Undergraduate Curriculum Committee

FROM: Prof. Ronaldo Menezes,
Program Chair, Computer Science and Software Engineering

CC: Dr. Marco Carvalho, Dean, School of Computing

DATE: October 20, 2016

RE: Adding a course
CSE 4242 – Advanced Algorithms
CSE 4231 – Computer Networks
CSE 4683 – Formal Methods

The School of Computing proposes the introduction of a new course called *Advanced Algorithms*. This course will eventually replace the CSE 4081 called *Introduction to Analysis of Algorithms*. No program is currently being affected by the creation of this course.

The other two courses, *Computer Networks* and *Formal Methods* are also being added. These classes also do not affect any of our programs.
Florida Institute of Technology

ADDING A NEW COURSE TO THE CURRICULUM

New courses are available beginning with the fall term in which they appear in the University Catalog.

This is a request for reactivation of a course in the system.  Yes ☐ No ☐

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>C</th>
<th>S</th>
<th>E</th>
<th>COURSE NO.*</th>
<th>4</th>
<th>2</th>
<th>3</th>
<th>1</th>
<th>CREDIT HOURS</th>
<th>3</th>
<th>ACADEMIC YEAR TO BE ADDED TO THE FILE</th>
<th>Fall 2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e.g., CSE)</td>
<td>(e.g., 1301)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(e.g., Fall 2010)</td>
</tr>
</tbody>
</table>

Justify level if 1000-level+ and no co- or prerequisites.

CLASS HOURS 45/semester  LECTURE HOURS 45/semester  LAB HOURS 0/semester  CONTACT HOURS (CEU ONLY) NA

DEPARTMENT Computer Sciences
(Specialized Sciences)

SCHEDULE TYPE Lecture
(e.g., Lecture, Lab or Special Topics/Project)

☐ COLLEGE OF AERONAUTICS – 23
☐ NATHAN M. BISK COLLEGE OF BUSINESS – 24
☐ COLLEGE OF ENGINEERING – 1
☐ COLLEGE OF PSYCHOLOGY AND LIBERAL ARTS – 25
☐ COLLEGE OF SCIENCE – 26
☐ EXTENDED STUDIES/NMB COLLEGE OF BUSINESS – 90
☐ SCHOOL OF COMPUTING – 29
☐ SCHOOL OF HUMAN-CENTERED DESIGN, INNOVATION & ART – 28

COMPUTER TITLE Computer Networks

This course will be entered into the system as: Bi-Level ☐ Cross-Listed ☐ Dual-Numbered ☐ Full-Load ☐ None of these/Standard Listing ☑

CATALOG TITLE Computer Networks

CATALOG DESCRIPTION OF COURSE Restricted to 25 characters, including spaces

Provides an overview of computer networks, applications and protocols. Includes network architectures, routing, addressing, medium access control, connection-oriented and connectionless services, network performance analysis, network programming and common transport-layer, network-layer and link-layer protocols.

This description has been approved by the catalog office

Emcj0y 10/20/2016

In addition, please attach a course syllabus and/or more detailed description.

<table>
<thead>
<tr>
<th>RESTRICTIONS</th>
<th>PRerequisite</th>
<th>CSE 2010</th>
<th>Corequisite</th>
<th>Course Number</th>
<th>and</th>
<th>or</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GRADERS TO BE ISSUED

☐ A, B, C, D, F
☐ A, B, C, D, F, CEU/Audit
☐ CEU
☐ S, U
☐ P, F
☐ Other

ADDITIONAL RESTRICTIONS ☐ and ☐ or (e.g., Major, Class Level, Department Head Approval)

If this course replaces a course currently offered in BANNER, please indicate old course information and the date/term the course may be removed from the system.

SUBJECT Alpha Prefix (e.g., CSE) __________ COURSE NO. (e.g., 1301) __________ TERM TO INACTIVATE __________

☐ Yes ☐ No Will this course be used to measure program-level student learning outcomes? If yes, review and signature required.**

☐ Yes ☐ No Will this course be used to satisfy the scholarly inquiry requirement? If yes, attach "Q" materials for review.

☐ Yes ☐ No Will this course impact any existing programs? If yes, attach "Changing Graduation Requirements" form for each program impacted.

APPROVALS: On completion of description and course number verification, affix appropriate signatures as indicated, and submit to the Office of Graduate Programs, or Undergraduate Curriculum Committee Chair for placement on agenda.

Originate 10/21/16

Chair, Graduate Council Date

Chair, Undergraduate Curriculum Committee Date

**Chief Academic Officer

Date

CREDIT & CATALOG MANAGER

These changes/additions have been made for the University Catalog and entered into the BANNER term named above.

Catalog & Curriculum Manager

REGISTRAR'S USE ONLY

SCACRESE  SCADETL  SCAPREQ  SCABASE  ACATALOG  SCARRIES  CIP Code  Operator Init.  Date

Florida Institute of Technology • Office of the Registrar

130 West University Boulevard, Melbourne, FL 32901-6975 • (321) 674-8114 • Fax (321) 674-7827

RGR-297-316
CSE 4231 - Computer Networks (3 credits)

Instructor: William Allen, PhD

Course Information:

Catalog Description: Overview of computer networks, applications and protocols, including network architectures, routing, addressing, medium access control, connection-oriented and connectionless services, network performance analysis, network programming and common transport-layer, network-layer and link-layer protocols.

Textbook:


Prerequisite: CSE 2010, Algorithms and Data Structures

Expected Outcomes: By the end of the course, students will have knowledge of the architecture of computer networks and the functions and protocols associated with the different layers of current networks. They will be able to capture and analyze network traffic and write software that interfaces with networked devices.

More specifically:

- Knowledge of the conceptual foundations of computer network and layered protocol architecture.
- Knowledge of different types of computer networks, such as WANs, LANs, wireless networks, and circuit-packet-switched networks, and between different paradigms of network applications (peer-to-peer/client-server).
- Understanding of the Internet architecture and the TCP/IP protocol suite, and details of representative protocols at the application, transport network and data link layers.
- Ability to use current network programming technology.

Course Topics:

- Computer Networks and the Internet (1 week)
- Application Layer (1 week)
- Transport Layer (3 weeks)
- The Network Layer: Data Plane (3 weeks)
- The Network Layer: Control Plane (3 weeks)
- The Link Layer (3 weeks)
- Wireless and Mobile Networks (1 week)
Grading Policy:

- Homework and Programming Assignments: 45%
- Class and online participation: 5%
- Midterm Exam: 25%
- Final Exam: 25%

Grading Scale:

A: 90 to 100, B: 80 to 89, C: 70 to 79, D: 60 to 69, F: below 60

Bibliography:

**Florida Institute of Technology**  

**ADDING A NEW COURSE TO THE CURRICULUM**

This is a request for reactivation of a course in the system.  
☐ Yes  ☐ No

**New courses are available beginning with the fall term in which they appear in the University Catalog.**

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>CSE</th>
<th>COURSE NO.*</th>
<th>4</th>
<th>2</th>
<th>4</th>
<th>2</th>
<th>CREDIT HOURS</th>
<th>3</th>
<th>ACADEMIC YEAR TO BE ADDED TO THE FILE</th>
<th>Fall 2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e.g., CSE)</td>
<td>(e.g., 1301)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(e.g., Fall 2010)</td>
<td></td>
</tr>
</tbody>
</table>

*Justify level if 1000-level+ and no co- or prerequisites.

<table>
<thead>
<tr>
<th>CLASS HOURS</th>
<th>45/semester</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LECTURE HOURS</th>
<th>45/semester</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LAB HOURS</th>
<th>0/semester</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CONTACT HOURS (CEU ONLY)</th>
<th>NA</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DEPARTMENT</th>
<th>School of Computing</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>(e.g., Biological Sciences)</th>
<th>(e.g., Lecture, Lab or Special Topics/Project)</th>
</tr>
</thead>
</table>

☐ COLLEGE OF AERONAUTICS – 23  
☐ NATHAN M. BISK COLLEGE OF BUSINESS – 24  
☐ COLLEGE OF ENGINEERING – 1  
☐ COLLEGE OF PSYCHOLOGY AND LIBERAL ARTS – 25  

<table>
<thead>
<tr>
<th>COMPUTER TITLE</th>
<th>Advanced Algorithms</th>
</tr>
</thead>
</table>

Restricted to 25 characters, including spaces

This course will be entered into the system as:  
☐ Bi-Level  ☐ Cross-Listed  ☐ Dual-Numbered  ☐ Full-Load  ☐ None of these/Standard Listing

<table>
<thead>
<tr>
<th>CATALOG TITLE</th>
<th>Advanced Algorithms</th>
</tr>
</thead>
</table>

**CATALOG DESCRIPTION OF COURSE** Restricted to 350 characters, including spaces

Introduces students to advanced techniques for the design and analysis of algorithms. Includes network flows, advanced data structures, linear programming, intractability, dealing with large datasets and computational geometry.

This description has been approved by the catalog office  

<table>
<thead>
<tr>
<th>Catalog &amp; Curriculum Manager</th>
<th>Date</th>
</tr>
</thead>
</table>

In addition, please attach a course syllabus and/or more detailed description.

**RESTRICTIONS**  
☐ Prerequisite CSE 2010  ☐ Corequisite No  and ☐ or

<table>
<thead>
<tr>
<th>Course Number</th>
<th>Course Number</th>
</tr>
</thead>
</table>

☐ Prerequisite  ☐ Corequisite  and ☐ or

<table>
<thead>
<tr>
<th>Course Number</th>
<th>Course Number</th>
</tr>
</thead>
</table>

☐ Prerequisite  ☐ Corequisite  and ☐ or

<table>
<thead>
<tr>
<th>Course Number</th>
<th>Course Number</th>
</tr>
</thead>
</table>

☐ Prerequisite  ☐ Corequisite  and ☐ or

<table>
<thead>
<tr>
<th>Course Number</th>
<th>Course Number</th>
</tr>
</thead>
</table>

GRADES TO BE ISSUED  
☐ A, B, C, D, F  ☐ A, B, C, D, F, CEU/Audit  ☐ CEU  ☐ S, U  ☐ P, F  ☐ Other

**ADDITIONAL RESTRICTION** ☐ and ☐ or

(e.g., Major, Class Level, Department Head Approval)

If this course replaces a course currently offered in BANNER, please indicate old course information and the date/term the course may be removed from the system.

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>Alpha Prefix (e.g., CSE)</th>
<th>COURSE NO.</th>
<th>(e.g., 1301)</th>
<th>TERM TO INACTIVATE</th>
</tr>
</thead>
</table>

☐ Yes ☐ No  Will this course be used to measure program-level student learning outcomes? If yes, review and signature required.**

☐ Yes ☐ No  Will this course be used to satisfy the scholarly inquiry requirement? If yes, attach "Q" materials for review.

☐ Yes ☐ No  Will this course impact any existing programs? If yes, attach "Changing Graduation Requirements" form for each program impacted.

**APPROVALS:** On completion of description and course number verification, affix appropriate signatures as indicated, and submit to the Office of Graduate Programs, or Undergraduate Curriculum Committee Chair for placement on agenda.

<table>
<thead>
<tr>
<th>Originator</th>
<th>Date</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Department Head/Program Chair</th>
<th>Date</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Dean or Associate Dean</th>
<th>Date</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Chair, Graduate Council</th>
<th>Date</th>
</tr>
</thead>
</table>

OR

<table>
<thead>
<tr>
<th>Chair, Undergraduate Curriculum Committee</th>
<th>Date</th>
</tr>
</thead>
</table>

**Chief Academic Officer**

<table>
<thead>
<tr>
<th>CATALOG &amp; CURRICULUM MANAGER</th>
<th>Date</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>REGISTRAR’S USE ONLY</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SCACORE</th>
<th>SCADETL</th>
<th>SCAPREQ</th>
<th>SCABASE</th>
<th>ACATALOG</th>
<th>SCASTRIES</th>
<th>CIP Code</th>
<th>Operator Init.</th>
<th>Date</th>
</tr>
</thead>
</table>

Florida Institute of Technology • Office of the Registrar  
150 West University Boulevard, Melbourne, FL 32901-6975 • (321) 674-8114 • Fax (321) 674-8227

RGR-297-816
ADVANCED ALGORITHMS, CSE 4242, Fall 2017
Instructor: Dr. William Shoaff
Text: Algorithm Design, Jon Kleinberg & Éva Tardos

Course Information: This is a senior-level course on the design and analysis of algorithms, covering several advanced topics not studied in typical introductory courses on algorithms. This course assumes "mathematical maturity" - that you are familiar with proofs, basic mathematical reasoning, run-time analysis of algorithms, etc.

Expected Outcomes: By the end of the course, students will have gained knowledge of different methods and techniques for designing advanced algorithms with real-world applications, will be able to critically analyze the pros and cons of applying these techniques in different contexts, and will be aware of many applications that require such techniques. Finally, students will be able to conceptualize and design efficient and effective end-to-end algorithmic solutions for different real-world problems.

More specifically:
- Students should develop a sound theoretical understanding of advanced algorithms and practical problem solving skills using them.
- Students should develop basic knowledge of a wide range of advanced algorithm design techniques including dynamic programming, linear programming, approximation algorithms, and randomized algorithms.
- Students should develop basic advanced algorithm analysis skills for analyzing the approximation ratio of approximation algorithms and the probability of randomized algorithms.
- Students should gain a good understanding on a wide range of advanced algorithmic problems, their relations and variants, and application to real-world problems.
- Students should understand design choices regarding the use of techniques such as Divide and Conquer, Greedy, and Dynamic Programming

Course Schedule:
- Week 1: Basic Algorithm Analysis
- Week 2-3: Graphs
- Week 4-6: Divide and Conquer, Greedy Algorithms, Dynamic Programming
- MIDTERM EXAM 1
- Week 7: Network Flow
- Week 8-9: NP and Computational Intractability
- Week 10: Approximation Algorithms
- Week 11: Randomized Algorithms
- MIDTERM EXAM 2
- Week 12: Local Search
- Week 13: Computational Geometry
- Week 14: Streaming Algorithms
• Week 15: External Memory

FINAL EXAM

Grading Policy: The average grade will be calculated based on 5-6 implementation of problems sets (35% total), 2 midterms (30% total), a final exam (30%), participation in class, in office hours, and Canvas (5%).

Bibliography:

Florida Institute of Technology

ADDING A NEW COURSE TO THE CURRICULUM

This is a request for reactivation of a course in the system.  □ Yes  □ No

New courses are available beginning with the fall term in which they appear in the University Catalog.

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>COURSE NO.</th>
<th>CREDIT HOURS</th>
<th>ACADEMIC YEAR TO BE ADDED TO THE FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C S E</td>
<td>4 6 8 3</td>
<td>3</td>
<td>Fall 2017 (e.g., 1301)</td>
</tr>
</tbody>
</table>

*Justify level if 1000-level and no co- or prerequisites

<table>
<thead>
<tr>
<th>CLASS HOURS</th>
<th>LECTURE HOURS</th>
<th>LAB HOURS</th>
<th>CONTACT HOURS (CEU ONLY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>45/semester</td>
<td>45/semester</td>
<td>0/semester</td>
<td>NA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPARTMENT</th>
<th>SCHEDULE TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>School of Computing</td>
<td>Lecture (A)</td>
</tr>
</tbody>
</table>

(e.g., Biological Sciences) (e.g., Lecture, Lab or Special Topics/Project)

□ COLLEGE OF AERONAUTICS – 23
□ NATHAN M. RISK COLLEGE OF BUSINESS – 24
□ COLLEGE OF ENGINEERING – 1
□ COLLEGE OF PSYCHOLOGY AND LIBERAL ARTS – 25

□ COLLEGE OF SCIENCE – 26
□ EXTENDED STUDIES/AMB COLLEGE OF BUSINESS – 90
□ SCHOOL OF COMPUTING – 29
□ SCHOOL OF HUMAN-CENTERED DESIGN, INNOVATION & ART – 28

COMPUTER TITLE: Formal Methods

Restricted to 25 characters, including spaces

This course will be entered into the system as:  □ Bi-Level  □ Cross-Listed  □ Dual-Numbered  □ Full-Load  □ None of these/Standard Listing

CATHOLIC TITLE: Formal Methods

CATALOG DESCRIPTION OF COURSE

Restricted to 350 characters, including spaces

Introduces the use of mathematical models of software systems for their specification and validation. Includes finite state machine models, models of concurrent systems, verification of models, and limitations of these techniques.

This description has been approved by the catalog office  Eby Nov 10/20/2016

Catalog & Curriculum Manager

Date

In addition, please attach a course syllabus and/or more detailed description.

<table>
<thead>
<tr>
<th>RESTRICTIONS</th>
<th>Course Number</th>
<th>Course Number</th>
<th>Course Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ Prerequisite CSE 3241</td>
<td>□ Corequisite Course Number</td>
<td>□ and □ or</td>
<td></td>
</tr>
<tr>
<td>□ Prerequisite CSE 4083</td>
<td>□ Corequisite Course Number</td>
<td>□ and □ or</td>
<td></td>
</tr>
<tr>
<td>□ Prerequisite Course Number</td>
<td>□ Corequisite Course Number</td>
<td>□ and □ or</td>
<td></td>
</tr>
</tbody>
</table>

GRAD nTo BE ISSUED
□ A, B, C, D, F
□ A, B, C, D, F, CEU/Audit
□ CEU
□ S, U
□ P, F
□ Other

□ Additional Restriction □ and □ or

(e.g., Major, Class Level, Department Head Approval)

If this course replaces a course currently offered in BANNER, please indicate old course information and the date/term the course may be removed from the system.

□ Yes  □ No  Will this course be used to measure program-level student learning outcomes? If yes, review and signature required.**

□ Yes  □ No  Will this course be used to satisfy the scholarly inquiry requirement? If yes, attach “O” materials for review.

□ Yes  □ No  Will this course impact any existing programs? If yes, attach “Changing Graduation Requirements” form for each program impacted.

APPROVALS: On completion of description and course number verification, affix appropriate signatures as indicated, and submit to the Office of Graduate Programs, or Undergraduate Curriculum Committee Chair for placement on agenda.

Originator  Date

Department Head/Program Chair  Date

Dean or Associate Dean  Date

Chair, Graduate Council  Date

OR

Chair, Undergraduate Curriculum Committee  Date

**Chief Academic Officer  Date

CATALOG & CURRICULUM MANAGER

These changes/additions have been made for the University Catalog and entered into the BANNER term named above.

SCADCRE  SCADTEI  SCAPRNG  SCABASE  ACALOG

SCARRES  CIP Code  Operator Init.  Date

Florida Institute of Technology • Office of the Registrar

150 West University Boulevard, Melbourne, FL 32901-6975 • (321) 674-8114 • Fax (321) 674-7827

RGR-297 816
CSE4683: Formal Methods

1 Catalog description

Official FIT Course description. Introduction to the use of mathematical models of software systems for their specification and validation. Topics include finite state machine models, models of concurrent systems, verification of models, and limitations of these techniques.

Prerequisites: CSE3421 [Software Design Methods], [and] CSE4083 [Formal Languages & Automata].

2 Textbook

The textbook: Understanding formal methods by Jean-François Monin [25]

3 Introduction

Formal Methods is a general term for system design, analysis, and implementation methods that are described and used with mathematical rigor. The purpose is to construct, with high confidence, systems that behave according to their specification. The two most important styles of formal methods for reasoning about software: model checking and deductive verification.

On the model checking side, the advantage of an automated method at the same time places restrictions on the kind of properties that can be verified. Accordingly, the verification of safety properties is key.

For the deductive verification side, there are many tools to choose from. KeY is an integrated tool for the formal specification and verification of Java programs. The tool supports formal specification in the Java Modeling
Language (JML), and translation from JML into logic. An interactive theorem prover is used to formally verify statements about specifications and programs.

4 Learning Objectives

The goals of the course are to:

- understand the potential and limitations of using logic based verification methods for assessing and improving software correctness,
- differentiate between syntactic, semantic, and proof methods in connection with logic-based systems for verification,
- express safety properties of (concurrent) programs in a formal way,
- describe the basics of verifying safety properties via model checking,
- use tools which integrate and automate the model checking of safety properties,
- write formal specifications of object-oriented system units, using the concepts of method contracts and class invariants,
- describe how the connection between programs and formal specifications can be represented in a program logic,
- verify functional properties of simple Java programs with a verification tool,
- acknowledge the socio-economical costs caused by faulty software, and
- approach the issue of correctly functioning software by means of abstraction, modeling, and rigorous reasoning

5 Topics Covered

- Introduction: Mathematical Preliminaries, first-order logic, formal languages. [one week = 3 hours]
- Concurrency: Theory of concurrency and causality, language for distributed and concurrent programming, dynamic logic, process algebra. [two weeks = 6 hours]

- Declarative Programming: Declarative approaches to search, non-monotonic logic. [two weeks = 6 hours]

- Model Checking: Property checking, predicate abstraction, automatic verification, reachability, state-space exploration. [two weeks = 6 hours]

- Security: Model checking, constraint solving. [two weeks = 6 hours]

- Java Modeling Language (JML). [one week = 3 hours]

- Predicate Logic for Specification of Java Programs. [two weeks = 6 hours]

- Translating JML into Dynamic Logic. [one week = 3 hours]

- Verifying Proof Obligations: Theorem Proving, proof development environments. [two weeks = 6 hours]

6 Tools

There is a wide range of tools used in the specifying, modeling, and verification of software. It is reasonable to expect that in the future, a different set of tools will be used to illustrate the concepts. We list some of these tools here.

1. Alloy, a modeling language simple but expressive logic based on the notion of relations, and was inspired by the Z specification language and Tarski's relational calculus. The Alloy Analyzer works by reduction to SAT [22].


3. Agda [8]

4. HOL [12]
5. KeY, a tool for the formal specification and verification of Java programs. The tool supports formal specification in the Java Modeling Language (JML). The core of the system is a theorem prover for the first-order dynamic logic [2].

6. Larch, a family of formal specification languages are intended for the precise specification of computing systems based on algebraic specification and abstract data types, [16]

7. SAL, one of the SRI FormalWare tools [4]

8. SPIN ("Simple Promela Interpreter"), general tool for verifying the correctness of distributed software models in a rigorous and mostly automated fashion. Systems to be verified are described in Promela (Process Meta Language), which models asynchronous distributed algorithms as non-deterministic automata and properties are expressed as Linear Temporal Logic [19]


10. Z [28]

7 Example Projects

Though the projects may vary from semester to semester, several large, group projects will be required.

Example Project A modeling project suitable for use with Z: “Terminator 3: A toy air traffic controller” from [13].

Example Project The classic dining philosophers shared resource problem is specification with a Promela models, safety properties expressed in temporal logic, and SPIN used to verify that none of the philosophers starve. (Adapted from a project at Chalmers.)
Example Project  Discover the security flaws of the Needham-Schroeder protocol[26] Specify contracts and invariants for a Java program in JML, verification of correctness with KeY. Apply the modifications to the protocol and proof that flaws are closed. (Adapted from a project at Chalmers.)

8  Grading

project one            15%
midterm exam one       15%
project two            15%
midterm exam two       15%
project three          15%
final exam             25%

9  Related Courses

These existing courses at other institutions were examined in order to create this course. The first course in the list, at Chalmers, was especially influential.

1. TDA293/DIT270 Software Engineering using Formal Methods at Chalmers University

2. CSSE 373 Formal Methods in Specification and Design at Rose-Hulman

3. 22C/55:181 Formal Methods in Software Engineering at the University of Iowa

4. CISC422/CMPE422/CISC835: Formal Methods in Software Engineering at Queen's University (Kingston, Canada).

References


