CIS-260 Software Specification and Design
Bristol Community College
Computer Information Systems Department
Spring 2020

Catalog Description

This course covers object-oriented analysis and design, methodologies and tools. It focuses on methodologies of specification and design of software systems. It addresses the issues of user interface design and software prototyping. The course also presents the state of the art in the tool and environments supporting the front end of the software development cycle.

- Three class hours and two lab hours per week.

Meeting days and times:

Laboratory: room K-105 Tuesdays and Thursdays 8:30 am – 9:20 am

Lecture: room K-105 Tuesdays and Thursdays 9:30 am – 10:45 am

Prerequisite:

CIS-158 or CIS 257 or permission of the instructor.

Instructor: Igor Kholodov Igor.Kholodov@bristolcc.edu

Office: K211

Telephone: 774-357-3328

Student Learning Outcomes

At the completion of this course the student will be able to:

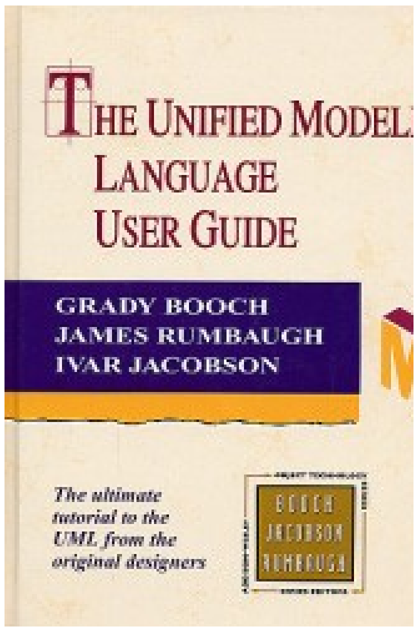
- learn about significance of modeling as a backbone of software development
- gain basic understanding of software process including quality management
- develop understanding and appreciation for formalism in software modeling and development
- learn Unified Modeling Language (UML) and its effective use in software development
- study wide range of design patterns and understand their role in software engineering

Specific goals to meet these outcomes include:

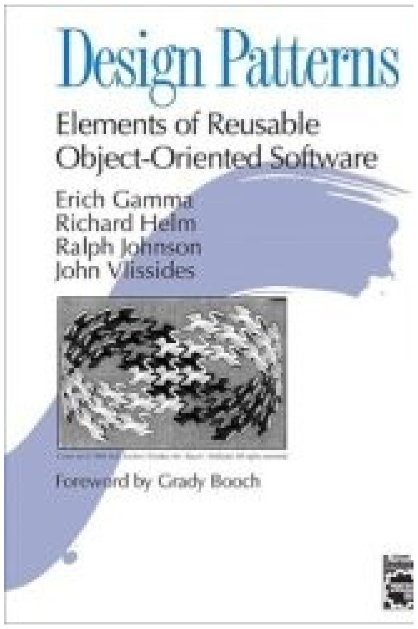
- study object paradigm in context of software design
- developing requirements analysis and software specification using object paradigm

- understanding the notion of design patterns in problem solving process
- experience working as part of a team on a moderate size design project
- meeting challenges involving oral and written presentation of technical information pertaining to the project

Recommended Texts



- Title: The Unified Modeling Language User Guide
- Publisher: Addison-Wesley Professional; Object Technology Series
- 2nd edition
- Hardcover: 496 pages
- ISBN-10: 0321267974
- ISBN-13: 978-0321267979



- Design Patterns: Elements of Reusable Object-Oriented Software
- by Ralph Johnson Richard Helm John M. Vlissides Erich Gamma
- Publisher: Addison-Wesley Professional Computing Series
- ISBN-10: 0201633612
- ISBN-13: 978-0-201633-61-0

Disability Accommodations

I encourage any student in need of accommodations for a specific documented disability to meet with me and the Office of Disability Services (508) 678-2811 (Fall River, ext. 2955; Attleboro and Taunton, ext. 2996; New Bedford, ext. 2955 and/or 4011) at your earliest convenience to ensure timely and appropriate accommodations. You may also contact the Office of Disability Services (ODS) online at http://www.bristolcc.edu/about/publicrecordsrequest/disabilityservices/

Course Organization

There will be two tests: Midterm, and Endterm.

The major component of this course is a **semester-long project** to be done in small groups (ideally 2 students per group). The schedule of project activities and the deliverables are going to be provided with the set of requirements during the second week of classes. Each group of students will have 2-3 scheduled meetings with the instructor for project review activities.

Each group will give a short (10-15 minute) presentation during one of the in-class project review sessions.

There will be weekly programming laboratories - attendance in labs is mandatory. The labs will start the second week of classes. The labs are designed to help you with the semester-long project. In the laboratories you will practice the use of UML diagrams in modeling requirements and system functionality.

Minimum Requirements for a Passing Grade

- Curiosity and love for learning.
- Responsibility for reading the textbook and keeping current with the class material.
- Ability to do research any subject not covered in text or study guides.
- Average grade of 60 or greater on Midterm and Endterm exams.
- Completion of all design stages of the semester-long project.

Weights for the Final Grade Determination

Weekly Quizzes	35%
Midterm Exam	20%
Endterm Exam	20%
Semester Project	25%

The Final Grades will be assigned as follows:

97-100	A+	87-89	B+	77-79	C+	67-69	D+
93-96	A	83-86	B	73-76	C	63-66	D
90-92	A-	80-82	B-	70-72	C-	60-62	D-
Below 60	F						

Teaching Methodology

- The lecture will be the principal teaching method used in this course. "Handouts" and sample programs will be available on the class web page. Youtube video notes are posted for the key concepts presented in this course. Class discussions will be conducted pertaining to the lab exercises. Software demos and overhead slides will be used.
- It will be imperative that the student complete all assigned readings and homework assignments *prior to class*. Failure to do so is a formula for failure. Coming to class prepared is essential for successful completion of the course.

Attendance Policy

Attendance is recorded weekly based on the student's ability to submit quality and timely lab/programming assignments each week. Students are considered "present" for the week if they submit the required lab assignment (with a satisfying passing grade) prior to the due date for that week. Poor attendance may affect your final grade.

Students are responsible for withdrawing officially if they stop attending any or all classes. Faculty no longer have the ability to withdraw a student from a class. A grade of "F" will be assigned to any student who stops attending a course but does not officially withdraw. Students are encouraged to meet with an advisor before making any changes to their schedule. Withdrawals impact Satisfactory Academic Progress and can place the student at risk for academic probation or dismissal. Students who use financial aid and who subsequently withdraw may be required to return some or all funds received. Withdrawals are accepted until the tenth week of classes. Students may withdraw online in accessBCC, in person at any Enrollment Center, or via their college email to enrollmentservices@bristolcc.edu. Email requests must come from the student's BCC college email address and must include the student's name, BCC student ID number, and course information (CRN, course and section number).

Email from non-college accounts will not be accepted. If a student officially withdraws after the third week of classes, there will be no tuition or college fee refunds. For more information, see the College Catalog at

http://bristolcc.edu/

Students with questions should contact Enrollment Services via any of the methods mentioned above or at 774-357-2590

BCC Academic Policies

College-wide Academic Policies outlined in BCC Academic Catalog directly apply to this course. It is your responsibility to read carefully and understand Academic Information, especially Academic Integrity, Academic Dishonesty, Academic Negligence, Plagiarism, and Classroom conduct, which are published online.

CIS-260 TENTATIVE SCHEDULE

Week 1. Intro to Software Specification and Design

- Reading: BCC Student Handbook, page 8, Tip #5:
    - *"For each hour in class, you should expect to study at least 2-3 hours outside of class. Know your limits, avoid over scheduling yourself (whether it be work or class). Set up a schedule that you know will allow you to earn good grades. And, maintain a day planner to help you stay organized."*

- Java Skills Review:

    Intro to Java Swing Forms (presentation)
    NetBeans GUI Builder, controls, and events (presentation)
    Event Listeners, Layout Managers, Event Handling (presentation)
- Sample code: CIS-257 Java Samples
- Semester-long review project, Point of Sale Support System.
- UML, the Unified Modeling Language.
- Design Models.

Weeks 2-3. Object paradigm top-down

- Presentation 1: Object-Oriented Paradigm (lecture).
- Object-oriented programming vs. object-oriented analysis and design
- Abstraction, encapsulation, information hiding, modularity
- Modeling as an engineering activity
- Modeling vs. prototyping

Weeks 4-5. UML, Unified Modeling Language

- UML in context of your Java experience
- Milestone: End of Week 4 -- Join the development team.
- OMG and its role in standardization process for object technology
- Power and limitations of modeling with UML

Weeks 6-7. Requirements Model

- Presentation 2: UML Use Cases and Requirements Modeling (lecture).
- Functional vs. non-functional requirements
- Domain and requirements scope
- Use case model
- Use cases and actors
- Priority in requirements - primary and secondary use cases and actors
- Modeling relationships between requirements - inclusion, extension, generalization

- Requirements analysis as a sub-process of software process
    - Involving *customer* and *user*
    - Conducting requirements review
    - Managing requirements
- Modeling use cases using sequence diagrams and activity diagrams
- Milestone: End of Week 7 -- Requirements model is due (use case model.)

Week 8. Requirements Model Review Sessions

- 30-minute sessions strictly by schedule.
- Sign-up in advance -- available times will be posted on the instructor's door.

Week 8. Analysis Model - Structure

- Moving from procedural paradigm of *functional requirements* (use-case model) to object-paradigm of *analysis model*.
- Presentation 3: UML Class Diagrams (lecture).
- Classes and types of classes
    - Class and class specification
    - Utility and template classes, template instantiation
    - Abstract classes and interfaces
- Relationships
    - Significance of relationships as the *mortar* of the software architecture
    - Inheritance - medium for expressing and controlling abstraction
    - Association, aggregation by reference, aggregation by value
    - Dependency
    - Direction and multiplicity of relationships
    - Naming relationships and roles
- Spring Break

Week 9. Analysis Model - Designing Objects with Responsibilities

- *Divide and concur* - making decisions in design process.
- Presentation 4: From Requirements to Classes (lecture).

    - Coupling and cohesion as opposite *forces* defining relationships
    - Synergy between defining modules and intra-module connections
    - Module (package, class) quality

End of Week 9. Midterm Exam, Theater Box Office Ticket Sales System

- End of Week 9 -- *Midterm*
- Construct **use case model**.
- Specify **primary actors**.
- Identify use cases for primary actors.
- Create **activity diagram** for a particular use case (with all **extensions** and **includes**.)
- Explain **types of UML relationships** between classes:
    1. Association
    2. Dependency
    3. Generalization/inheritance
    4. Aggregation by value
    5. Aggregation by reference
- Provide examples that illustrate various relationships.

Week 10. Analysis Model - Behavior

- Presentation 5: UML Sequence and State Diagrams (lecture).
- Modeling dynamic properties of the system - spectrum of the *tools*
- Sequence diagrams in modeling interaction between modules
- Modeling behavior of a module using state diagram
- Modeling concurrency with activity diagrams
- Activity diagram with states
- Role of formalism in modeling
    - State diagrams - automata
    - Activity diagrams - Petri Nets
    - Generating executable code from a model
    - Executable specification

    - Factors in choosing the level of formalism
- Milestone: End of Week 10 -- Analysis model is due.

Week 11. Analysis Model Review Sessions

- Week 11
- 30-minute sessions strictly by schedule.
- Sign-up in advance -- available times will be posted on the instructor's door.

Week 12. Design model

- Moving from analysis to design
- Role of non-functional requirements in design
- Considering constraints and properties of implementation environment
- Design model as a refinement of analysis model in context of implementation environment
    - Defining Subsystems
    - Design classes and relationships
    - Design and plan iterations
    - Where does the design end?
- Milestone: End of Week 13 -- Design model is due.

Week 13. Implementation model

- Implementation and unit testing
- Role of use cases as test cases
- Subsystem integration

Week 14. Design Patterns

- Definition of design pattern
- Using design pattern as a reusable solution in problem solving process
    - Defining problem constraints and evaluating applicability of a known pattern to the problem at hand
    - Essential components of design pattern to make it a reusable asset
- Domain independent design patterns

    - Creational Patterns
    - Structural Patterns
    - Behavioral Patterns
- Domain specific design patterns
- Automation of patterns definition and reuse in IDEs

End of Week 14. Endterm Exam, Car Rental System

- End of Week 14 -- *Endterm*
- Construct detailed state diagram with all states and transitions.
- Provide activity diagram for a particular use case (with all extensions and includes)
- Describe the Pool of Resources pattern (N-ton.)
- Describe the Proxy pattern.
    - Construct class diagram for the Proxy pattern.
    - Define and comment all classes and interfaces.
- Explain software extendability issues.

Week 15. Semester Project Review

- End of Week 14 -- Entire project is due
- Week 15 -- Entire project review sessions.
- 30-minute sessions strictly by schedule.
- Sign-up in advance -- available times will be posted on the instructor's door.

Summary of Presentations and Video Notes

- INTRODUCTION TO SWING AND OOP
    - Week 1: .Java Skills Review
        - Intro to Java Swing Forms (presentation)
        - NetBeans GUI Builder, controls, and events (presentation)
        - Event Listeners, Layout Managers, Event Handling (presentation)
    - Weeks 2-5: Object-Oriented Paradigm (lecture)

        - Introduction: The World of Objects (hi-rez WMV 9 MB.)
        - Abstraction Mechanisms (hi-rez WMV 10 MB.)
        - UML Interfaces (hi-rez WMV 12 MB.)
        - Modeling as Engineering Activity hi-rez only, WMV 16 MB.
- REQUIREMENTS MODEL
    - Weeks 6-7: UML Use Cases and Requirements Modeling (lecture)
        - Case Study: Point-of-Sale System (hi-rez WMV 8 MB.)
        - Requirements Modeling and Use Cases, Part I (hi-rez WMV 6 MB.)
        - Requirements Modeling and Use Cases, Part II (hi-rez WMV 10 MB.)
        - An Overview of Project Stages and Design Models
        - Elaborate Use Case Example, Part I (hi-rez WMV 9 MB.)
        - Elaborate Use Case Example, Part II (hi-rez WMV 10 MB.)
        - UML Use Case Diagrams (hi-rez WMV 11 MB.)
- BUSINESS ANALYSIS MODEL - STRUCTURE
- Week 8: UML Class Diagrams (lecture)
    - Adding Associations to Class Diagrams (hi-rez WMV 11 MB.)
        - UML Class Diagram Association Roles and Multiplicity (hi-rez WMV 5 MB.)
        - UML Class Diagram, Adding Class Attributes (hi-rez WMV 5 MB.)
        - UML Class Diagram, Data Types, and Attributes (hi-rez WMV 11 MB.)
    - Week 9: From Requirements to Classes (lecture)
        - UML From Requirements to Classes. Part I (hi-rez WMV 9.5 MB.)
        - UML From Requirements to Classes. Part II (hi-rez WMV 9.5 MB.)
        - UML Class Quality Part I: Designing Objects with Responsibilities (hi-rez WMV 9 MB.)
        - UML Class Quality Part II: Designing Objects with Responsibilities (hi-rez WMV 10 MB.)
        - UML Class Quality Part III: Designing Objects with Responsibilities (hi-rez WMV 10 MB.)
        - UML Class Quality Part IV: Designing Objects with Responsibilities (hi-rez WMV 10 MB.)

    - UML Class Quality Part V: Designing Objects with Responsibilities (hi-rez WMV 9.7 MB.)
- BUSINESS ANALYSIS MODEL - BEHAVIOR
  - Weeks 10-14: UML Sequence and State Diagrams (lecture)
    - UML Sequence Diagrams (hi-rez WMV 6 MB.)
    - Sequence Diagrams and Use-Case Realizations
- DESIGN MODEL
  - Design Model and Class Diagrams
  - Runtime Behavior and State Diagrams
- IMPLEMENTATION MODEL
  - Implementation Model and Writing Code
  - Design Patterns (lecture)
  - Object Pool Pattern (lecture)

Note: *This syllabus is a suggested course outline and will be generally followed, subject to change according to the instructor's discretion and needs. Academic flexibility is important.*