Syllabus CS 682

Course Description

MET CS 682
Information Systems Analysis and Design

This course describes modern methods of information system analysis and design for organizations with IT resources. It introduces the discovery process for system feasibility, describes stakeholder analysis, and covers requirements analysis. The course explains use cases and their application to requirements analysis. It covers the management of system analysis projects and risks. "Build vs. buy" trade-offs are discussed. The Unified Modeling Language for specifying object-oriented system designs is discussed. Data flow diagrams and activity models are integrated with the analysis and design coverage. The course covers most of the fundamental system architectures, as well as approaches to detailed design.

Course Objectives and Learning Goals

This course is designed to enable you to do the following

- Discriminate among types of business systems
- Explain and summarize a proposed systems analysis project
- Assess and critique the issues of working in a team
- Distinguish between system-level and low-level requirements
- Distinguish between architectural and detailed designs
- Rank the goals of good system design
- Understand UML models

By reading the lectures and completing the assignments in this course, you will be able to:

- Better predict and deal with risks
- Plan and design a project schedule
- Develop written functional and non-functional requirements
- Create written use cases and scenarios
- Integrate the use of classes in Object-Orientation
• Relate one class to another through inheritance, aggregation and association
• Create sequence diagrams and other UML diagrams
• Construct system architectures and detailed designs

**Week-by-week Topics**

**Week 1 — Introduction and Process**

• Types of business systems
• Participants in systems analysis
• A systems analysis example
• Introduction to system process
• Development process alternatives
• Requirements, design and quality assurance
• Configuration management

**Week 2 — System Development Processes, Risk and System Design Trade-offs**

• System development project management
• Formal and agile processes, their advantages and disadvantages
• Team inter-personnel issues
• Risk management
• Project scheduling
• Organizational structures
• Legacy applications
• Agile approaches
• Team Software Process

**Week 3 — System and Requirements Analysis**

• The meaning of “requirements”
• System-level requirements
• Detailed requirements
• Functional requirements
• Non-functional requirements
• Techniques for interviewing and documenting requirements
• Introduction to design of user interfaces
• Introduction to use cases, data flow diagrams, state transition diagrams

Week 4 — Modeling with UML

• Classes
• Class relationships
• More on use cases
• An example of using UML
• Sequence diagrams
• State models
• Activity diagrams

Week 5 — System Architectures

• Design purposes
• Software frameworks
• More on data flow diagrams
• ATAM Design and Tradeoffs
• Categorizing system architectures
• Component technology

Week 6 — Object-Oriented Designs

• Design in the Unified Development Process
• Designing against component interfaces
• Specifying classes and functions for design
• Software reuse
• Detailed sequence diagrams and data flow diagrams
• Software reuse
• Standards for detailed design
• Estimating cost of software

Week 7 — Final Exam

Weekly Activities

Each week you will need to:

• Read the online lectures
• Read recommended pages in the textbook (listed below)
• Complete the interim assessment for interim feedback
• Complete the homework assignment(s)

Instructor Biography

Eric Braude received his Ph. D. from Columbia University in mathematics and Master's in Computer Science from the University of Miami. He taught at CUNY and Penn State, followed by twelve years in government and industry as a software engineer, scientist, and manager. He is an Associate Professor of Computer Science at Boston University’s Metropolitan College where he has at times held the chairmanship and the acting associate deanship. His research concerns reliable program construction. Eric has written, co-written, or edited six books, including “Software Engineering” and “Software Design.”

(For a complete resume, see http://www.bu.edu/csmet/files/2012/04/Eric-Braude-Resume-2012.pdf)

Computer Science Department
Metropolitan College
Boston University
808 Commonwealth Ave Room 258
Boston, MA 02215

http://www.bu.edu/csmet/braude/

Initial Course Development

This course was originally developed by Professor Eric Braude.
Weekly deliverables and due dates for Discussions, Assignments, and Assessments will be supplied when the course begins.

Module 1 Study Guide and Deliverables

Readings: Online lectures
Whitten & Bentley, Primary: 6–16, and 30–33. Secondary: pages 4–65

Module 2 Study Guide and Deliverables

Readings: Online lectures

Module 3 Study Guide and Deliverables

Readings: Online lectures

Module 4 Study Guide and Deliverables

Readings: Online lectures
Secondary: Chapters 9 and 10
Module 5 Study Guide and Deliverables

Readings: Online lectures

Whitten & Bentley: Primary: pages 445-467 (Most of this material is not covered in the notes) Secondary: 468–515 (This material serves as backup to, and gives another perspective on the topics in these notes)

Module 6 Study Guide and Deliverables

Readings: Online lectures

Whitten & Bentley, pages 646–679

Course Resources

Required Course Textbook


Textbook Notes

- Make very sure that you are getting the 7th edition.
- The textbook for this course can be purchased from Barnes & Noble at Boston University.
- McGraw-Hill/Irwin provides an online learning center associated with this text. It does not replace the textbook. Your assignments for this course will be based on the printed version of the textbook. However, you might find it useful to review the case studies, practice interim assessments and PowerPoint presentations available for each chapter of the textbook.

**Supplemental Material**

- You will find a section with supplemental material on the CS 682 Online Campus course homepage.

**Other Resources**

- For definitions and terms, and for pointing you to references, Wikipedia can sometimes be useful. However, remember that information at Wikipedia is erratically curated, and entries have been manipulated by a variety of people for a variety of reasons. You are free to use Wikipedia as a starting point and as a source of pointers to higher-quality information, but avoid citing Wikipedia (or similar sources that have not been reviewed professionally for veracity) as authorities.
- The UML specifications are at [www.omg.org/technology/documents/formal/uml.htm](http://www.omg.org/technology/documents/formal/uml.htm) (but you will find them very dense and formal indeed).
- We will use Visio in this course for UML. However, you are free to use other tools if you wish.

**Boston University Library Link**

As Boston University students you have full access to the BU Library—even if you do not live in Boston. From any computer, you can gain access to anything at the library that is electronically formatted. To connect to the library use the link [http://www.bu.edu/library](http://www.bu.edu/library). You may use the library's content whether you are connected through your online course or not, by confirming your status as a BU community member using your Kerberos password.

Once in the library system, you can use the links under “Resources” and “Collections” to find databases, eJournals, and eBooks, as well as search the library by subject. Some other useful links include:

Go to [http://www.bu.edu/library/research/collections](http://www.bu.edu/library/research/collections) to access eBooks and eJournals directly.

If you have questions about library resources, go to [http://www.bu.edu/library/help/ask-a-librarian](http://www.bu.edu/library/help/ask-a-librarian) to email the library or use the live chat feature.
To locate course eReserves, go to http://www.bu.edu/library/services/reserves.

Please note that you are not to post attachments of the required or other readings in the water cooler or other areas of the course, as it is an infringement on copyright laws and department policy. All students have access to the library system and will need to develop research skills that include how to find articles through library systems and databases.

**Evaluation of Students and Grading**

Absorbing and creating IT perspectives will be expected of all students in the class. To attain excellence ("A" work), you will be expected to develop your own analyses and comparisons.

**Basis for Grades**

There are four components to your grades.

1. **Weekly Assignments**

   Most of the content of the course will be explored through weekly assignments that study actual cases or encourage you to extrapolate from your own organizations and experiences. These are counted equally.

2. **Interim Assessments**

   These are straightforward questions intended to help you with the weekly assignments and associated subject matter.

3. **Final**

   There will be a three-hour final exam which is similar in format to the homework's. This provides you the opportunity to show what you have learned from the material, the discussions, and from doing the homework.

**Grade Computations**

The course grade will be computed from the following:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly Assignments</td>
<td>60%</td>
</tr>
</tbody>
</table>
Interim Assessment | 10%
---|---
Proctored Final Exam | 30%

Evaluation Criteria and Grading Rubric

The project phases are graded according to the evaluation matrices on pages that follow. These are averaged using A=95, A-=90, B+=87, B=85, B-=80 etc.

To get an A grade for the course, your weighted average should be >93. A-=≥90. B+=≥87. B≥83. B-:=≥80 etc.

The interim assessment grades are Acceptably on track (1), Not yet acceptably on track (0). Otherwise:

≥5 " Acceptably...": A
≥4 " Acceptably...": B
≥3 " Acceptably...": C
=1 " Acceptably...": D
None "meets ...": F