|  |  |  |
| --- | --- | --- |
| ­ |  | ISM 6225 Distributed Information Systems |

SYLLABUS

**Office Hours** : 10 – 11 on Fridays and by appointment

**Pre-requisites** : One course in OOP & interest in developing distributed applications

**Teaching asst** : Details on Canvas

**Resources** : Visual Studio (Community edition recommended). If needed, please seek an appointment with the TA in the first week of class to complete the required setup.

**Textbooks** : (we will only use a few chapters from each book, but cover-to-cover walkthrough is recommended)

1. (AF) Pro ASP.NET Core 6 [electronic resource] / Adam Freeman, ISBN: 978-1-4842-7957-1 e-book available from the USF library (<https://link.springer.com/book/10.1007/978-1-4842-7957-1>)
2. (JL) Programming entity framework [electronic resource] : Code First / Julia Lerman & Rowan Miller, ISBN: 9781449323851 (electronic bk.), 1449323855 (electronic bk.), 9781449323844 (electronic bk.), 1449323847 (electronic bk.), e-book available from the USF library (<https://ebookcentral.proquest.com/lib/usf/detail.action?docID=806527&pq-origsite=primo>)
3. For students with a limited background in programming, I can recommend C# for Absolute Beginners, “Learn to Think Like a Programmer and Start Writing Code” (2nd Edition), 2021, Download from Library here: [https://link.springer.com/book/10.1007/978-1-4842-7147-6](https://nam04.safelinks.protection.outlook.com/?url=https%3A%2F%2Flink.springer.com%2Fbook%2F10.1007%2F978-1-4842-7147-6&data=05%7C02%7Cmagrawal%40usf.edu%7C494330cf7e1a4e69661308dc1168fee5%7C741bf7dee2e546df8d6782607df9deaa%7C0%7C0%7C638404389851108105%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=WwBROn4%2FW0ydkaluXhMzcC8Rf0W70YoCfBXIZ1OZPjg%3D&reserved=0)

**Optional books (used for the Javascript and OSI model sections, power point slides are available on Canvas)**

1. (AD) Business data communications and IT infrastructures (3rd Edition) / Manish Agrawal & Clinton Daniel
2. A smarter way to learn JavaScript, <http://www.asmarterwaytolearn.com/index.html>

**Course websites**:

Readings : <http://magrawal.myweb.usf.edu/dis/>

Code repository : <https://github.com/ISM6225>

# Course Objectives

End-user applications are the lifeblood of modern industry, and BAIS graduates are at the forefront of developing end-user applications, which also give them the opportunity to bring their data science models to a wide audience. This course introduces students to basic business application development using modern frameworks and toolsets, while showing how various technology components interact with each other to deliver a meaningful end user experience. Students will develop familiarity with using the cloud to deploy these applications. Projects and exercises will focus on helping students gain familiarity with these technologies. Students find the course challenging, but also rate it as a robust introduction to technology.

Every graduate course in every business school has one primary motivation: **why should anyone follow you?** The projects and exercises in the class are therefore designed to help students develop confidence in their own technical and leadership capabilities so they can inspire confidence in team members and clients about their ability to develop technology solutions to address business problems.

# Learning Outcomes[[1]](#footnote-1)[[2]](#footnote-2)

1. Reinforce competence with computer programming and object-oriented programming
2. Introduce concepts of algorithms and data structures
3. Develop proficiency in using html and css for web user interfaces
4. Develop proficiency in using common developer tools such as GitHub for version control
5. Introduce Java Script (JS) and JS frameworks for client-side user interaction
6. Introduce standard frameworks for full stack business application development
7. Introduce the use of cloud technologies to run web business applications
8. Understand networking technologies used to architect distributed systems from application components

Figure 1: ISM 6225 course module time allocations and objectives

# Logistics

1. The course may be seen as comprised of three modules – Figure 1 shows these modules and their respective time allocations. The first module covers the OSI model, the underlying networking technologies, which helps students architect distributed business applications appropriately. Module 2 reinforces basic programming concepts and introduces algorithmic thinking to solve standard computer programming problems. A benefit of this module is that it introduces students to the approaches adopted by technology intensive organizations in their interviews. Module 3 introduces basic web development, which is useful to help students get started with developing web applications that use data and analytical models.
2. All students should have access to an installation of Visual Studio. If needed, please seek an appointment with the TA in the first week of class to complete the required setup.
3. Faculty teaching this class (in particular, Prof. Clinton Daniel) are creating video walk-throughs to help students with the hands-on components. These videos are designed to be viewed and followed before class meetings so that class time can be used to help students overcome any technical challenges with implementing the concepts covered. Links to these videos are available on Canvas.
4. Web development is one of the core activities of the BAIS profession. For many students however, this class is their first introduction to full-stack business application development and even programming with a full complement of tools. It is recommended that you use this opportunity to develop competence and overcome any fears in this evergreen essential skill. This class is also an opportunity to begin building your GitHub portfolio.
5. Students may form groups to complete projects. Follow instructions on Canvas for group size. All students should sign up on the groups pre-created for the class, do not create new group sets. In case of group dysfunction, students may fire disruptive members, or sub-groups may create their own group at any time.
6. Please use the term “ISM 6225” in the subject line of your email (no spaces) to help me filter emails.
7. Readings and assignment deliverables are specified on the course site on Canvas.
8. Deliverables are due by the end of day on the due date (usually this means 11:55pm). The default due days are Sundays, but adaptations are made to accommodate calendar and scheduling issues.
9. Make up opportunities will only be provided for job-related situations and for medical emergencies in the immediate family.
10. The instructors will do their best to teach you something useful, not merely certify what you already know. This impacts exam preparation for example, where you should not expect a shortlist of questions to prepare.
11. The instructors conduct all sections of the course tightly in sync. Accordingly, students can reach out to any instructor of the course to expand the availability of office hours.
12. On all programming assignments, you are expected to handle all reasonable corner cases.
13. If your prior background has not prepared you well in technology (e.g. students from non-technology majors), it is your responsibility to devote the time to get up to speed with the content. Slowing down to accommodate your preferences takes away the value of the class for other students.

# Business Continuity

In the event of an emergency, USF may opt to continue delivery of instruction through methods that include but are not limited to: Canvas, Teams, and email messaging and/or an alternate schedule. It’s the responsibility of the student to monitor Canvas for each class for course specific communication, and the main USF, College, and department websites, emails, and MoBull messages for important general information.

# Presentation

To help build your communication skills and algorithmic thinking, each group will be required to pick a data structure or algorithm and present it to the class. Please use the instructions in the presentation assignment for details.

# Grading

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Default Type** | **Unit weight** | **Total weight** |
| Project and Presentation | Group | 10% | 10% |
| Programming problems | Individual | 1% | 5% |
| Exams | Individual | 18%, 20% | 38% |
| Assignments | Group/ Individual | 5%, 15% (I), 08%, 5% | 33% |
| Attendance | Individual | 11% | 11% |
| Feed forward | Individual | 3% | 3% |

# Grading Policy

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Total%** | **Grade** | **Total%** | **Grade** | **Total%** | **Grade** | **Total%** | **Grade** |
| >=95  >=90  >=87 | A+(max 10% of class)[[3]](#footnote-3)  A  A- | >=84  >=80  >=77 | B+  B  B- | >=74  >=70  >=67 | C+  C  C- | >=63  >=60  <60 | D+  D  F |

# Changelog (from Fall 2023)

1. Changed sequence of OpenKattis problems to distribute course activities evenly across the semester
2. Dropped API consumption
3. Dropped persistence layer
4. Dropped dictionaries
5. Dropped virtual machines

# Relevant readings (for students interested in greater background coverage)

## Technical

* + - 1. (CLRS) Cormen, Thomas H, Leiserson, Charles E, Rivest, Ronald L. and Stein, Clifford, Introduction to Algorithms, Third Edition, 2009, isbn 0262033844, The MIT Press (standard reading for top tier SDE interviews)
      2. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, Algorithms, 2006, <http://algorithmics.lsi.upc.edu/docs/Dasgupta-Papadimitriou-Vazirani.pdf> (easier read on algorithms)
      3. Kaufman, Charlie, Perlman, Radia and Speciner, Mike, Network Security: Private Communication in a Public World, Second Edition, 2002, isbn 9780137155880, Prentice Hall Press

## Business

1. Bilton, Nick. Hatching Twitter: A True Story of Money, Power, Friendship, and Betrayal. New York, N.Y.: Portfolio/Penquin, 2013.(available in the USF library)

# USF Core Syllabus Guidelines

Details are available on the University’s Core Syllabus Policy Statements page: <https://www.usf.edu/provost/faculty/core-syllabus-policy-statements.aspx>

# References

David Gooblar, “The missing course: Everything They Never Taught You about College Teaching,” Harvard University Press, 2010

# Useful Links

Interviewing: <https://twitter.com/siddharthm83/status/1447220066043318285?s=20>

**ISM 6225: Tentative course outline[[4]](#footnote-4)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week | Module | Lecture topic | Class activities | Deadlines |
| 1 | Module 1 – OSI Model | Syllabus, OSI Model |  | Attendance, group formation  Programming problems 1 – OpenKattis intro |
| 2 | OSI Model (contd.), Network security, support services | Wireshark | Programming problems 2 – OpenKattis Level 1.3 |
| 3 | Module 2 – programming introduction and computational problem solving | IDE introduction, Programming Introduction, source control | Visual Studio walk-through, Git profile URL with fork project and teamwork | Assignment 1 – OSI Model (Group) |
| 4 | Programming introduction (contd.) | Time complexity | Programming problems 3 – OpenKattis Level 1.4 |
| 5 | Object-oriented programming concepts introduction, inheritance | Simple OOP program | **Exam 1 (module 1) in BSN 1100**  **11:30am – 12:30pm on Saturday, March 9** |
| 6 | Module 3 – Full stack business application development | HTML, CSS | MyWeb deploy | Programming problems 4 – OpenKattis Level 1.5 |
| 7 | JavaScript, charting | Simple chart | Programming problems 5 OpenKattis Level 1.3 (JavaScript) |
| 8 | MVC components – controllers, action methods, views, models, model binding, routing | Simple MVC site | Assignment 2 – Computational problem solving (Individual) |
| 9 | MVC components – controllers, action methods, views, models, model binding, routing (contd.) | Simple view with links, Model binding | **EXAM 2 (programming intro, OOP) in BSN 1100; 11:30am – 1pm on Sat, April 13** |
| 10 | MVC misc - Form validation, cloud deployment | Azure Deploy | Assignment 3 – UI Look and Feel (Group) |
| 11 | MVC Project Presentations | In-Class Group Presentations | Assignment 4 – JavaScript and Charting (Group)  MVC Project – Deployed to Azure |

1. https://teaching.cornell.edu/teaching-resources/designing-your-course/setting-learning-outcomes [↑](#footnote-ref-1)
2. https://www.depts.ttu.edu/opa/resources/docs/Writing\_Learning\_Outcomes\_Handbook3.pdf [↑](#footnote-ref-2)
3. At instructor’s discretion, among the top students in the class [↑](#footnote-ref-3)
4. In the interests of the class, deviations may be made in the coverage of topics as outlined in the tentative course calendar. However, to help plan your calendars for the rest of the semester, assessment and deadline dates will be non-negotiable after the first day of class. Some deadlines may be clubbed together to accommodate schedules. [↑](#footnote-ref-4)