



Computer-based Decision Support Systems (OPIM 557)

University:	Georgetown
School:	McDonough School of Business
Department:	Operations and Information Management (OPIM)
Course:	Computer-based Decision Support Systems (557)
Section:	41 - Monday nights from 6:30pm to 9:20pm
Credits:	1.5
Prerequisites:	N/A

Description

This course explores advanced topics in management support systems with a focus on decision theory. The course will present insights into key issues and problems in decision support information systems. The Visual Basic for Applications (VBA) programming language within MS Excel will be used as the primary tool to build basic decision support systems. Class deliverables will be systems built with VBA code.

Learning Objectives

1. Discuss topics related to information systems, with focus on decision-support systems.
2. Create decision-support systems in MS Excel, using native interface elements and the VBA programming language.
3. Gain marketable programming and data skills.
4. Be confident in your ability to learn any programming language.
5. Have fun!

Community

Students

This course is a graduate business school (MBA) elective, with a maximum enrollment of 50 students per section.

Professor

Adjunct Professor Michael Rossetti, a professional data scientist and software developer, will be teaching this course. Students should feel free to direct questions to the professor by sending a Slack direct message to [@prof-rossetti](#) or an email to mjr300@georgetown.edu. If emailing, all parties should use university-issued addresses. The professor aims to respond to messages within around one to two business days.

When sending announcements and replying to students, the professor may send messages outside of normal business hours. There is no expectation for students to keep the same schedule. Students should feel free to read and reply to messages at whatever time is most preferable for them!

The professor will hold office hours at least once per week (see Canvas Calendar), as well as by appointment.

Materials

Texts

Students are encouraged to consult online resources such as:

- [VBA Language Reference](#) (Microsoft.com)
- [Excel VBA Reference](#) (Microsoft.com)
- [Learning VBA in Excel](#) (Lynda.com)
- [Intro to VBA in MS Excel](#) (Prof Rossetti)

Students may also consult texts such as:

- [Excel VBA Programming For Dummies](#), by John Walkenbach
- [Microsoft Excel 2016 Power Programming with VBA](#), by Michael Alexander

Further Reading:

- [The Lean Startup](#), by Eric Ries
- [Rework](#), by Jason Fried and David Heinemeier Hansson
- [The Design of Everyday Things](#), by Don Norman

Computers

Each student should have access to a personal portable computer during class. The computer should allow use of MS Excel, as well as the Visual Basic for Applications (VBA) programming language and ActiveX Controls found inside of MS Excel. Students are encouraged to use a Windows operating system because MS Excel for Mac may not include ActiveX Controls. Any student who doesn't have access to a suitable computer during class may inquire with the technology center about [loaning a laptop](#) or using virtualization software.

Operations

Canvas

All registered students should have access to the [Canvas](#) learning management platform. The course calendar in Canvas is the most up-to-date source of information about the scheduling of class sessions, office hours, and deliverable due dates. Students will be expected to submit deliverables through Canvas except when otherwise instructed, and the professor will distribute all grades through the Canvas gradebook.

GitHub

GitHub is the leading online platform for sharing software and code-related resources. The course [GitHub repository](#) is the primary source of course materials, including programming language references, instructional exercises, and project descriptions. The course repository may also include a copy of all email announcements after they are sent.

Slack

Slack is a chat platform that will be used to share code snippets, links to helpful materials, and other incidental course communications. All students should join the [OPIM 557 Slack organization](#) at the beginning of the semester when invited by the professor. Students are encouraged to post questions and answers in the [#section-41](#) channel and to monitor the [#section-41-videos](#) channel for links to class recordings. Students may optionally also join the [#course-materials](#) channel to subscribe to a news feed of updates to the course repository and the [#events](#) channel to discuss upcoming events and opportunities for industry networking. The professor may create additional channels as applicable to serve assignment-specific purposes or facilitate group communications.

Reference: [OPIM 557 Slack Onboarding](#), [Emoji Cheat Sheet](#) 😊

Evaluation

Student learning will be evaluated through submission of an onboarding survey and six weekly progress check-ins (15% total), four hands-on programming projects (60% total), and a final exam (25%). Students should consult the Canvas Calendar for due dates and weights of all deliverables.

The professor aims to provide grades for all submitted items within around seven to ten days after the due date, and may utilize graduate assistants during the grading process. Any student who has a question or concern about a grade should ask the professor in writing within seven days of receiving the grade, and the professor will look into the matter in a timely manner.

Extra Credit

Additionally, the professor may award extra credit “Engagement Points” to be applied towards the final exam. Activities eligible for engagement points include, but are not limited to:

- Contributing meaningfully to classroom discussions.
- Volunteering programming solutions and strategies during class.
- Displaying meaningful effort and progress in class and/or during office hours.
- Helping other classmates (in “reasonable” ways) in class, during group office hours, or in the course Slack channel.
- Contributing new and/or improved class materials by proposing changes to the course repository and getting those changes accepted. Includes everything from fixing typos, to clarifying language, to adding context, to creating new reference documents altogether.
- Delivering projects which go above and beyond the minimum requirements in impressive and inspiring ways, while still meeting all of the minimum requirements.

Projects

Simple System

The *Simple System* acts as an introduction to information systems, software application development, programming with VBA in MS Excel, user interface design, and data management. Students will create an interactive GUI application which accepts user form inputs and saves corresponding records in a separate worksheet.

Retirement Savings Calculator

The *Retirement Savings Calculator*, a Decision-Support System, is a practical financial modeling tool. Students will create an interactive GUI application which combines user inputs and risk preferences, historical data, and probability, to predict how long retirement savings will last under various scenarios.

Executive Dashboard

The *Executive Dashboard*, a Management Information System, provides business reporting capabilities to aid decision-making. Students will create a tool which automates the process of transforming monthly sales data into a summary report of business insights, including the aggregation of total sales and identification of top-selling products. The dashboard will utilize charts and graphs to help tell a compelling story.

Stock Trading Recommendation System

The *Stock Trading Recommendation System*, a Decision-Support System, processes real-time market data through a custom recommendation algorithm to provide investors with a competitive advantage. Students will create an interactive GUI application which fetches historical trading prices from the Internet, and combines them with user risk preferences to recommend whether or not to purchase a given stock or cryptocurrency.

Schedule

The schedule below is tentative and may change to reflect actual pace of instruction. The Canvas Calendar will reflect the most up-to-date scheduling information, including due dates.

Date	No.	Focus
Mar-18	1A	<ul style="list-style-type: none"> • <u>Topics</u>: Information Systems, Decision-Support Systems, Process Automation, Object-oriented Programming with VBA in MS Excel • <u>VBA Programming</u>: Getting Started, <i>Macros</i> • <u>Exercise</u>: “Hello Macro”
	1B	<ul style="list-style-type: none"> • <u>MS Excel Objects</u>: <i>Workbooks, Worksheets, Ranges</i> • <u>Interface Elements</u>: <i>Message Boxes, Input Boxes, Command Buttons</i> • <u>VBA Programming</u>: <i>Sub-procedures, Workbook Events</i> • <u>Exercise</u>: “Smart Workbooks”
Mar-25	2A	<ul style="list-style-type: none"> • <u>Topics</u>: User Experience and Interface Design • <u>Interface Elements</u>: <i>Scroll Bars, Combo Boxes, etc.</i> • <u>VBA Programming</u>: Variables, Basic Datatypes (<i>Strings, Numbers, Booleans</i>) • <u>Exercise</u>: “All the Controls”
	2B	<ul style="list-style-type: none"> • <u>Interface Elements</u>: <i>Check Boxes, Option Buttons, etc.</i> • <u>VBA Programming</u>: Conditional Logic and Control Flow (<i>If statements</i>) • <u>Exercise</u>: “All the Controls” (cont’d)
Apr-01	3A	<ul style="list-style-type: none"> • <u>Topics</u>: Code Simplification, System Maintenance, Quality Control, Tests • <u>VBA Programming</u>: <i>Functions, Modules</i> • <u>Exercise</u>: “Codebase Cleanup”
	3B	<ul style="list-style-type: none"> • <u>Interface Elements</u>: <i>User Forms</i> • <u>Project</u>: “Simple System”
Apr-08	4A	<ul style="list-style-type: none"> • <u>VBA Programming</u>: Iteration (<i>Loops, Arrays</i>) • <u>Exercise</u>: “Processing CSV Strings”

	4B	<ul style="list-style-type: none"> • <u>Topics</u>: Probability, Risk, Uncertainty • <u>Project</u>: “Retirement Savings Calculator”
Apr-15	5A	<ul style="list-style-type: none"> • <u>VBA Programming</u>: Workbook File Operations • <u>Exercise</u>: “Processing Spreadsheet Files”
	5B	<ul style="list-style-type: none"> • <u>Topics</u>: Metrics, Reporting, Data Visualization • <u>MS Excel Objects</u>: <i>Pivot Tables, Charts</i> • <u>Project</u>: “Executive Dashboard”
Apr-22	N/A	Easter Break (no class)
Apr-29	6A	<ul style="list-style-type: none"> • <u>Topics</u>: Computer Networks, the Internet, and HTTP • <u>VBA Programming</u>: <i>WinHTTP Services</i> • <u>Exercise</u>: “Processing Internet Data”
	6B	<ul style="list-style-type: none"> • <u>Topics</u>: Application Programming Interfaces (APIs) • <u>Project</u>: “Robo Advisor”
May-07	N/A	<ul style="list-style-type: none"> • Final Exam

Policies

Attendance

All students are encouraged but not required to attend class in-person. If not able to attend class in-person, students are still expected to review the assigned course materials, view the audiovisual class recordings, stay apprised of the schedule of deliverables, and participate in remote communications in Slack as applicable.

Instructional Continuity

If for any reason a class session is not able to be held in-person (e.g. due to inclement weather) or the professor is not able to attend in-person (e.g. due to illness), the professor will announce a specific instructional continuity plan, which may involve remote instruction or Graduate Assistant-led instruction.

Late Submissions and Extensions

Late submissions are generally not accepted. However there are a few exceptions to this rule.

First, although deliverables are generally due by 11:59pm on their respective due date, if a deliverable is submitted after the deadline but before an instructor has begun the grading process (e.g. a project submitted at 2am before the instructor wakes up around 10am to start grading the next day), the instructor is encouraged to consider that submission as being “on-time” and to include it in the list of deliverables to evaluate.

Second, students may request an “Advanced Notice Due Date Extension” by emailing the professor at least 72 hours in advance of the original due date. In their request, students should describe the reason for their request and propose an alternative due date. Example reasons include work travel, interviews, and family obligations. Students should expect to submit deliverables on time unless the professor explicitly approves their extension request in writing, in which case an alternative due date will be agreed upon and late penalties will be waived.

Finally, students may request a “Short Notice Due Date Extension” by emailing the professor anytime within the 72 hours immediately preceding the due date. In their request, students should describe the reason for their request and propose an alternative due date as well as a proposed late penalty. Example reasons include time mismanagement and last-minute emergencies. Students can expect to have their due date extension approved, and depending on the specific timeline and circumstances of the request, a modest late penalty may be applied. This option is to be used as an academically honest alternative to what would otherwise be a temptation to commit an academically dishonest action (see academic integrity section below).

Final Grades

The Dean's office has mandated the maximum mean final grade in this course should be 3.5 (between A- and B+). Further guidance from the Dean's office:

What constitutes an A, A-, B+, etc. is determined by each individual faculty member. Unlike most undergraduate programs, there is not a standard scale of numeric to letter grades. For example, 94% and above doesn't necessarily equate to an A, 90-93% doesn't necessarily equate to an A-, etc. Generally faculty will look at the final numeric grades as a whole and look for natural breaks.

Learning Accommodations

Any student requiring learning accommodations, such as longer exam periods, must register and coordinate through the university's [Academic Resource Center](#).

Code of Conduct

Students should abide by all policies set forth by the university's [Office of Student Conduct](#).

Academic Integrity

Students are expected to follow the university's [Honor System](#) and [Graduate School Academic Policies](#), as well as those set forth here. For progress check-ins and the exam, student collaboration is strictly prohibited. However for project deliverables, some collaboration is natural and helpful. The following guidelines define the nature of acceptable collaboration on project deliverables. They have been adapted from [Harvard CS50's Academic Honesty Policy](#), which centers around examples of “reasonable” vs. “unreasonable” activities.

To summarize:

1. You may help others, but you may not do their work for them.
2. You may ask others for help, but you may not ask them to do your work for you.
3. Your work product must materially originate from you and you alone (except in some “reasonable” exceptions noted below, which must in all cases be appropriately cited and attributed).
4. If receiving help in “reasonable” ways noted below, you must accompany any resulting code with proper attribution citations (i.e. code comments including URLs or email addresses to specifically identify the helpful information sources). Citations are also encouraged, but not required, for information originating directly from class instruction, course materials, or the official programming language documentation. Prefer to over-attribute than under-attribute.

Below are non-comprehensive lists of example activities which can be considered “reasonable” or “not reasonable”, respectively. Any further questions about what constitutes an academic integrity infraction should be proactively directed to the professor. Instructors are advised to report any suspected violation of academic integrity, and violations may lead to serious consequences such as failure or dismissal.

Reasonable / Acceptable

- OK - Discussing concepts, strategies, and techniques in a human language, in pseudocode, or via illustrations and diagrams.
- OK - Accessing and/or sharing links to publicly-available online resources, including documentation, tutorials, blog posts, video walkthroughs, etc., as well as course materials, exams, and project descriptions, even from past semesters.
- OK - Asking someone else for help debugging your program by sharing a few lines of your code with them or running your code in their presence.
- OK - Helping someone else debug their code by providing general guidance, sharing links to public resources, and/or suggesting or sharing a snippet or few lines of code.

Unreasonable / Violations

- NO - Accessing deliverables submitted privately by other students, past or present.
- NO - Basing one's own deliverable materially off of any other, including but not limited to submissions from past semesters, even if attribution is present.
- NO - Decompiling, deobfuscating, or disassembling the work product of any other student, past or present, even if attribution is present.
- NO - Failing to properly attribute any line of code originating from a source which requires attribution (see "Reasonable" guidance above).
- NO - Sharing more than a few lines of code from your solution with a classmate if they need help. This includes a prohibition on sending, receiving, downloading, or otherwise sharing entire files of code.

Acknowledgement and Authorization

Class sessions will be recorded and distributed back to students via a university-issued platform called Panopto. Students should be aware that audiovisual class recordings may include their image, name, and voice. Any student who would like to opt out of class recordings should email the professor within the first week of enrolling, and the professor will suggest some reasonable accommodations, which may include sitting in designated areas and adopting a different nickname during class discussions.