

# ISAT 340—Software Development

## Fall 2011 Syllabus

---

### Introduction

#### **You Are Your Own Best Teacher**

The way software gets developed changes constantly. If you don't stay on top of it, you will quickly find your skills out-of-date. Programming languages and styles of coding are constantly changing. The only way to stay current in this practice is to develop an understanding of where to look to find relevant tutorials and descriptions of how to do what you want to do. I'm happy to show you some of my favorites, but at the end of the day learning to find them on your own is the only reliable way to keep up to date. That being said, this course is not without structure.

### Overview

In this course, students will expand and enhance their knowledge of the processes, methods and techniques of efficient and effective software application development. Students will explore a variety of software models and examine the types of applications for which the models are most useful. Students will also learn how the various models can be integrated to produce a comprehensive software solution to a given problem.

### Course Objectives

The goal of this course is

- To familiarize you with the creation and enhancement of software systems in a sophisticated development environment
- To reinforce procedural (algorithmic) and object-oriented knowledge modeling (acquisition, representation, and programming)
- To introduce database systems and terminology and ASP.Net technology
- To practice teaming, project management, and analytical writing skills.

### Topics Worth Spending Time On

There is so much ground that could be covered in this course that we will only be able to scratch the surface of most topics. If you put significant time and energy into this course you should, however, be able to leave it with some marketable skills. Here are some of the topics that you may cover time permitting:

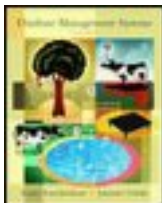
- Relational Database Development
  - Database Management System (DBMS) Software  
e.g. Microsoft Access, Microsoft SQLServer
  - Structured Query Language (SQL)

- Database Analysis—How to get information from clients
- Database Design—How to turn client-info into a specification
- Database Implementation—How to turn a specification into a database
- Database Integration—How to use the database in an application
- Web Application Development
  - *Server-side Web Languages*—How processing happens on the web  
e.g. ASP.Net
  - *Client-side Web Languages*—How web information is structured  
e.g. XHTML, Javascript (more depth on these in ISAT 348)
  - *Web Protocols*—How information travels around the web  
e.g. HTTP, FTP/SFTP, SSH, SMTP, RTSP, Jabber/AIM
  - *Web Servers*—How information is stored on the web  
e.g. IIS, SQLServer, Subversion
- Development Environments, e.g. Visual Studio
- Project Management Environments, e.g. Microsoft Project, Basecamp
- Web Application Genres  
e.g. E-commerce, Blog, Community/Social Networking, Project management, Content Management (CMS), Wiki
- Other topics (if time permits)
  - LINQ
  - Website analysis and planning
  - Proposal writing
  - Web service publishing and subscription
  - Mobile applications and design for handheld devices
  - Incorporating 3<sup>rd</sup> party code

Your instructor **loves** to learn about new topics in this field, so please, if you don't see something here that you want to cover, please let me know.

## Books

I recommend two books. One will help you learn about databases. The other will be a handy reference for programming in .Net.



[Ramakrishnan, Raghu \(1999\) Database Management Systems. McGraw-Hill Education, 0072322063](#)



[Gaddis, Tony and Irvine, Kip \(2010\) Starting Out with Visual Basic 2010, 5th ed. Addison Wesley, 0136113400](#)

## Software

You can download the necessary software for this course from the JMU MSDN Academic Alliance website. Namely, you can get Microsoft Visual Studio 2010, which is the programming environment we'll be using. You can also get a copy of Microsoft Access 2007, which we'll be using for database development. You should have received an email with your username and password, however you can recover this information from the website below:

[https://msdn03.e-academy.com/elms/Security/Login.aspx?campus=jmu\\_cs](https://msdn03.e-academy.com/elms/Security/Login.aspx?campus=jmu_cs)

## Schedule

### Weekly Use of Class Time

One of the keys to a successful class experience is to have a posted, regular and predictable schedule that everyone can follow. That way, nobody ever has to ask “what should I be doing now?” As such I propose the following use of our weekly class time:

Day	Activity
<b>Monday</b>	Lab time—there will be a new lab <i>every</i> week
<b>Wednesday</b>	Labs due by 9AM. Grade labs together in class.
<b>Friday</b>	Cover new topics, tie up loose ends, assign lab for next week
<b>Monday Night</b> (optional)	Hacking session, 8-Midnight, ISAT 337

### Out of Class Preparation

If you expect to get anything substantial out of this class, you need to make a commitment to spend time out of class figuring out how to do this stuff.

### How much time should I spend?

Of course, that's entirely up to you. The federal government considers a 12-hour schedule to be “full-time.” A full-time job is 40 hours/week, so **I recommend that at a minimum you commit to spending 10 hours per week** on this class, including time you spend in class and in hacking sessions. Of course, the subject matter is deep enough that you could easily spend much more time than this. I strongly recommend that you keep a log of all time you spend on this class.

### How should I spend my outside prep time?

Excellent question. I'm glad you asked. Here are some recommendations:

- Choose your own tasks
- Choose tasks that:
  - Are interesting to you
  - Require you to acquire skills/knowledge that **you** want to get
  - Have clearly defined beginning and end points, i.e. you should be able to cross it off of a checklist
  - Are not too ambitious, i.e. take at most a few hours to complete

- Match your preferred style of learning
- Make a plan—start early and set explicit tasks each week
- Keep a log of whether or not you completed them
- Take a few minutes each week to review your task completion

What types of tasks am I referring to? Here are some suggestions:

- Complete an online tutorial
- Watch a video on a particular topic or skill
- Read an article or chapter
- Tackle a project or lab to implement something
- Get help or tutoring from a classmate/TA/your professor/outside person
- Analyze/explore examples of websites or other multimedia artifacts you consider to be good
- Write your own tutorial or create your own tutorial video
- Come to the hacking session
- Stop by your prof's office to chat

For each topic we cover explicitly I will make some recommendations about specific chapters to read or tutorials to watch. I'll also create labs designed to give you experience with specific skills. However, this is just a beginning recommendation. Please augment my suggestions with your own!

### Rough Schedule of Topics

I make no guarantees that this is indeed an accurate list of topics, nor of the exact schedule by which we'll follow them, but just to give you an idea:

#### Week Date Topic(s)

<b>1</b>	8/29	Course Intro, Database Analysis
<b>2</b>	9/5	Database Design
<b>3</b>	9/12	Database Implementation
<b>4</b>	9/19	SQL
<b>5</b>	9/26	Database Integration
<b>6</b>	10/3	Intro to ASP.Net
<b>7</b>	10/10	XHTML
<b>8</b>	10/17	Form Handling
<b>9</b>	10/24	Databases in ASP.Net
<b>10</b>	10/31	Advanced ASP.Net
<b>11</b>	11/7	Web Services
<b>12</b>	11/14	LINQ
<b>13</b>	11/28	Mobile Apps
<b>14</b>	12/5	Wrap-Up
<b>15</b>	12/12	Exam/Presentations

## Important Dates

The following dates are for the Fall 2011 semester:

- Monday, August 29th: First Day of Class
- Tuesday, September 6<sup>th</sup>: Last day to Drop/Add without Department Head signature
- Thursday, September 15th: Last day to Add a course with Department Head signature
- Friday, September 16th: Last day to Withdraw from JMU with a full refund
- Monday-Friday, November 21st-25th: JMU Thanksgiving Break
- Friday, December 9th: Last Day of Class
- Monday, December 12th, 8:00AM: Section 2 Final Exam
- Friday, December 16<sup>th</sup>, 8:00AM: Section 1 Final Exam

## Weekly “Hacking” Sessions

From [Wikipedia](#):

*“Hackers follow a **spirit of creative playfulness** and **anti-authoritarianism**, and sometimes use this term to refer to people applying the same attitude to other fields.”*

There will be a weekly “hacking” session held in ISAT/CS room 337 on Mondays from 8pm to midnight. Your instructor and/or TA will be present during this session to work with you on anything class-related. These sessions are designed to be a **fun, relaxed, and very informal**. Feel free to come by get extra help on stuff you’re having trouble with, to work with your prof/TA on some of the projects they’re working on this semester, or perhaps to learn some “extra” tricks or secrets on how to be an effective programmer. This is definitely not required, but is surely destined to be the coolest Monday night hang out spot around so I’m sure you won’t want to miss it.

## Evaluation

### Simply Put: Grades Undermine Learning

The evidence to support this claim is ample, robust and has been growing for the past 40 years (cf. [Deci, Koestner, and Ryan 2001](#)). While it’s possible to use grades in a way that minimizes their negative influence on learning, rather than jump through those hoops, your instructor would like to do away with them altogether. It’s simpler, less work, and causes much less stress for everyone

involved with the process. Incidentally, it's also a much more effective way to foster learning.

### **But wait! Don't grades offer important performance feedback?**

Maybe, but they aren't really effective at offering feedback unless they are accompanied by robust comments, and if you've got robust comments, what really does the grade add?

### **Yeah, okay, but grades motivate me to study!**

Yes, that's exactly the problem. They aren't really doing what you think they're doing. [Watch this video clip from The Office.](#)

If you are genuinely interested in learning the material in this course, why do you need grades to motivate you? And if you aren't interested in the material covered in this course, why are you here? I mean really! Why would you pay money to do something you don't want to do? I don't have any desire whatsoever to force-feed this material down your throat. If you don't learn to become a software developer, it has little, if any, impact on me. I love the material and I do this job because I truly love sharing my passion with others. I can be a good guide to you but I have no interest in dragging you along against your will. What does either of us gain by that? If you still don't know yet, whether or not you like this stuff, I'm also happy to engage with you to help figure out the answer to that question. If you give the material an honest effort and find you still don't like it, that, in my opinion, is a valuable use of both yours and my time.

### **Yeah, but employers want people with good grades...**

Nope. Employers want independent problem solvers who know how to create and deploy effective software. They want people who know the difference between good work and crap—people who can figure out for themselves how to achieve desired outcomes. Why in the world would a potential employer care about the letter grade on a transcript if you've got a résumé chock full of real live software projects that you've completed? Even though grades are designed to be a proxy for showing what you can do, they are never as good as the real thing. Take advantage of your time in this class to build skills and products that will adequately show what you can do.

### **Okay, but at least the registrar wants a grade. What will you tell them?**

We will have a one-on-one discussion and decide this together. JMU doesn't force me to follow any sort of grading distribution or curve. JMU doesn't charge me \$100 per A, \$50 per B, \$25 per C and so on. It has little impact on me whatsoever if I give all A's or all F's. Personally, I think you should give yourself an honest grade that reflects what you can do. I will help you arrive at that determination. At the end of the semester you and I will have a one-on-one meeting in which we discuss everything you've done this semester and figure out together what to tell the registrar.

### **That doesn't really sound fair.**

Fair to whom? Concepts of fairness only really come into play in the context of a competition between people over scarce resources. We've already established that there's nothing scarce about the supply of A's in the world. The only competition in which there's a potential for unfairness is in competitions for things like jobs, internships, admission to graduate school, and scholarships. The truth is that none of these competitions are really 100% fair anyway, though they try to be. Furthermore, my way of distributing grades is, while unorthodox, no less valid than anyone else's method. For an accessible discussion on the validity of grades I refer you to [Alfie Kohn's 2002 article on grade inflation in the Chronicle of Higher Ed](#). When you really dig into the dank underbelly of how grades are distributed, you're going to find that most of us faculty are just making it up as we go along. Rather than bicker with you over something about which I don't care, my choice is just to let you choose for yourself what you want. Instead, I'll spend my time sharing really juicy content with you, and trying to find out what makes you tick.

### **Alright already, so if you don't give grades, how am I going to know if I'm doing a good job or not?**

Now we're talking.

The whole point of my approach is to make you forget about grades altogether and instead focus on learning how to produce high quality software. The first question is, what should we be studying? [Watch this video on current trends](#).

Okay, so this video is guilty of a bit of hyperbole and should be taken with a grain of salt, but in my mind here's the money quote:

*We are currently preparing students for jobs that don't yet exist, using technologies that haven't been invented, in order to solve problems we don't even know are problems.*

This goes double in a software development class. I've been programming for over 10 years now, and I've seen the state of technology and the practices of development change completely several times within that time frame. What that means is that we have to have a sense of humor about the topics on the syllabus. Although they serve as an interesting framework around which we can spend our time together, it would be somewhat naïve of us to really believe that knowing how to do any of the things there is really going to serve us well for more than three or four years. As such, I believe that a practical and rational set of goals for you to have when you take this course is:

1. Figure out if you really like building software
2. Figure out what your strengths and weaknesses are in this domain

If it turns out that you love doing this and are also likely to be good at it, you may be in what Sir Ken Robinson calls your Element. I think all of your "higher

education” should be about finding your element. But don’t let me convince you, [have a listen to Sir Ken](#).

So, if you take what Sir Ken says seriously (and I do), here’s what I think it means for us in this class. There are three areas in which your work in software development might be evaluated:

1. **Did you enjoy it?** I don’t think you need my help in figuring out whether or not you like this stuff. You should be able to evaluate that for yourself.
2. **Does it work?** Software pretty much works, or it doesn’t. Although we’ll talk about this some, I generally expect that you can decide this for yourself as well.
3. **Does it have style?** Under the surface however, there are some technical aspects of software quality that may only be apparent to a trained eye. I don’t expect you to have much experience these aspects and that’s where I can bring the greatest amount of my own expertise to bear in this class. This is where I see my efforts as having the greatest impact. My goal is to train you to be able to evaluate the technical aspects of software on your own.

At the end of the day, the responsibility for determining the quality of your work is your own and here’s why:

*If you can’t, on your own, tell the difference between high and low quality work, then you will never ever be able to reliably produce high quality work by yourself because...how would you know?*

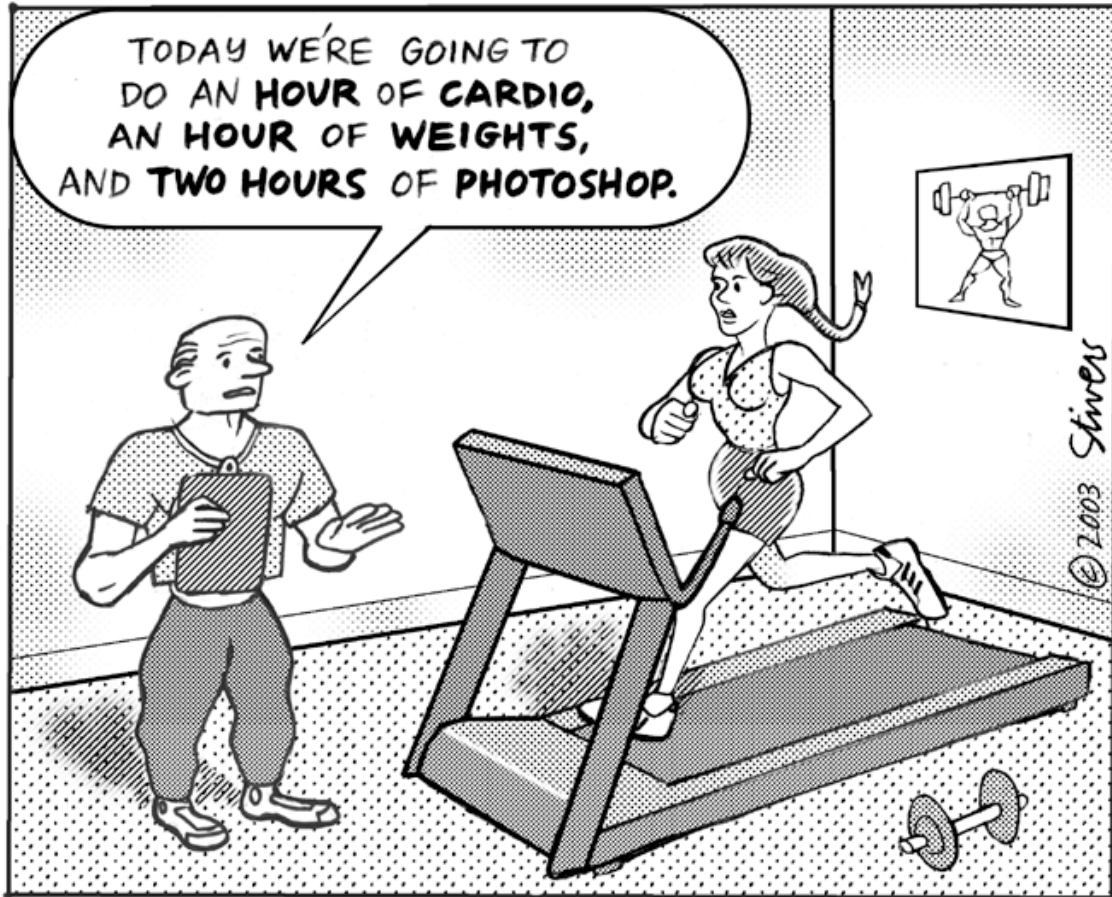
I see my role as your instructor as a guide and cheerleader to help you find your passion. Actually, I really see myself as more like your personal trainer.

So what do I mean by that?

1. You have hired me to help you get stronger at software development
2. I can show you what exercises to do, but you are the one who has to put in the time and do the heavy lifting
3. I still get paid, regardless of what you do, so why not get your money’s worth?

I really love this stuff, so why not put in everything you’ve got and get something out of what this class has to offer.





## Academic Integrity

Any breach of integrity will be grounds for immediate failure of the course. I take this very personally. However, I **strongly** encourage sharing with attribution. The [DRY principle \(Don't Repeat Yourself\)](#) is one of the most celebrated cultural elements in the programming community. This includes incorporating other people's code, where appropriate. Being able to use other people's code effectively is an extraordinarily valuable skill. So, in other words, I encourage collaboration, but make sure to attribute credit to those people who (knowingly or unknowingly) helped you out. The one caveat is I **strongly discourage you from copying without comprehension**. It would be pretty stupid to turn in a bunch of copied projects on which you learned absolutely nothing. Frequently code that you download from the web comes with explicit copyright preferences of the authors stated. Please respect them.

## Attendance Policy

Attendance is not mandatory, but strongly encouraged. This is *your* learning experience and I want you to make decisions about how to spend your time responsibly. I highly value the contribution you make not only to your own learning but also to your classmates as well as a member of a learning community. Therefore, I take attendance every class just to make sure I'm not losing anyone, but ultimately your grade depends upon what is in your portfolio.

## Other Policies

You can see other JMU policies regarding inclement weather, disability accommodations, and religious observances can be found at <http://www.jmu.edu/syllabus>

## Your Prof

Name: Morgan Benton

Office: HHS 3224

Office Hours: Mondays 2-4, or other times by appointment

Office Phone: 540 568 6876

Cell Phone: 973 495 7736 (calls and texts are ok within reason)

Calendar: my [Google calendar](#) is usually pretty accurate

Email: bentonmc@jmu.edu

Facebook: <http://www.facebook.com/morgan.benton>

Twitter: <http://twitter.com/morphatic>

Yahoo!/AIM: mcbenton17 (but I'm almost never logged in)

I'm a highly available person. I'm usually on campus 9-7 Mon-Sat. Tuesdays I shut my door to try to get some research done, so I'd appreciate your help in keeping that "me" time clear, but otherwise meeting with students is one of the most enjoyable parts of my job. You can check out my Google calendar above. If not otherwise indicated, I'm probably in my office.