ISAT 252--Programming & Problem-Solving--Syllabus

Spring 2018

Course Overview

Welcome to ISAT 252, Programming and Problem-Solving! This course introduces you to the basics of computer programming and also to the ways that programmers approach problems. "Computational Thinking" is a term coined by Jeanette Wing, Chair of the Computer Science Department at Carnegie-Mellon University. According to Dr. Wing, Computational Thinking:

- is a way of solving problems, designing systems, and understanding human behavior that has become a fundamental part of the way people think and understand the world.
- means creating and making use of different levels of abstraction, to understand and solve problems more effectively.
- means thinking algorithmically to develop more efficient, fair, and secure solutions.
- means understanding the consequences of scale, not only for reasons of efficiency but also for economic and social reasons.

The goal of this course is to give you a new set of tools that will expand your ability to solve real problems for real people.

Grades

Your grade this semester will be based upon a point system. You earn points by earning badges. Each badge has a certain number of points associated with it. The point cutoffs for various grades are in the table below. I will not round or curve. You are responsible for keeping yourself on track. To get an "A", you need to be earning 8-10 points a week, on average. There should be plenty of opportunities to earn points. If for some reason, you find yourself without anything to do, please take your prof out for coffee (you can earn a point that way), and we can come up with a plan for you. **Please see the important caveat to this grading system in the attendance policy below**.

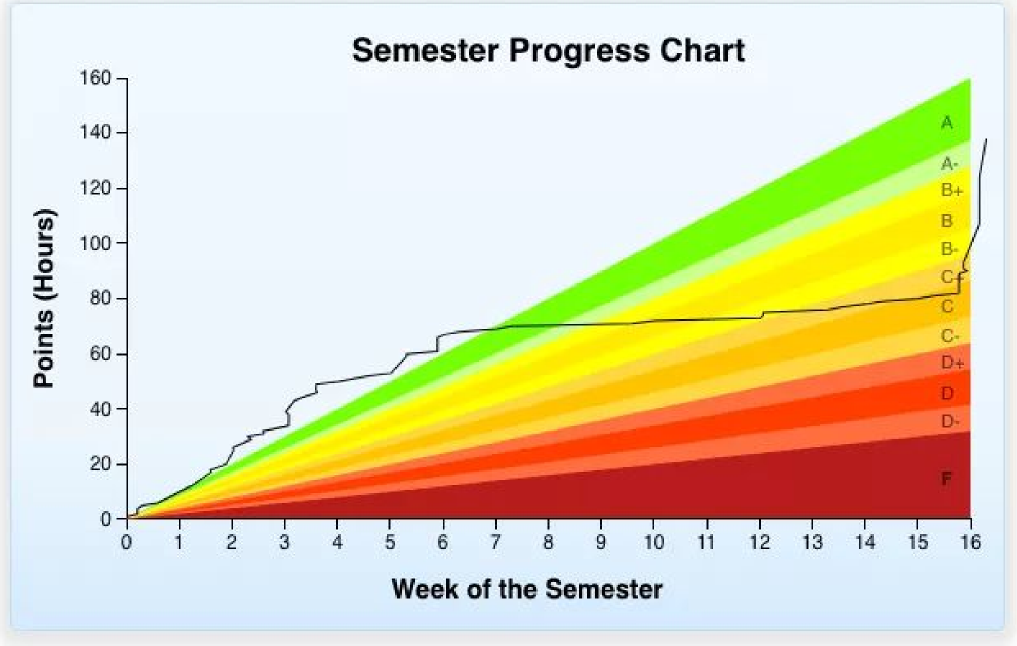
Grade	Points
A	129-150
A-	120-128
B+	111-119
B	99-110
B-	90-98
C+	81-89
C	69-80
C-	60-68
D+	51-59
D	39-50
D-	30-38
F	0-29

Attendance Policy

Attendance is mandatory. I will take attendance at the beginning of every class period. If you are not there when your name is called, you will not get credit for being there. If you are marked absent more than five (5) times during the course of the semester ***for any reason, excused OR unexcused*** your semester grade will drop by one letter. Woody Allen said, “80% of success is just showing up.” I care if you’re here.

Weekly Point Cap

VERY IMPORTANT: You can NOT earn more than 15 points in any single week. I’ve implemented this rule because last year, virtually all of my students had semester progress charts that looked like this:



This is the WRONG way to do this class!!!

In other words, they started out the semester very strong. Then around week six, they stopped doing any work at all. Then at the very end of the semester, they crammed in a bunch of work all at once. This was a problem for a number of reasons:

1. The quality of the work they did at the last minute was very poor
2. I'm pretty sure a lot of them turned in work that they didn't understand
3. As such, the quality of the learning was pretty low
4. Cramming in all of this half-assed work at the last minute is not very much fun
5. Having to grade everything all at once was really not fun for the instructor
6. It robbed the students of the opportunity to ask questions and really enjoy what they were doing
7. It robbed the instructor of the opportunity to spend time with people and really get to know them

So, the "official" week in this class begins at 4 AM on Mondays and goes until the following Monday at 4 AM. You cannot receive credit for more than 15 points during any one-week period. Please be mindful of this as you plan out your time this semester. This policy is designed to:

1. Encourage you to devote time to the class gradually and consistently over time, which research indicates is the BEST way to learn
2. Reduce your level of stress (and your instructor's) as we get towards the end of the semester

3. Prevent you from working too hard, or devoting too much time to this class in any one week

Textbook

This semester we will be reading a book by Charles Petzold called Code: The Hidden Language of Computer Hardware and Software. We will read the entire book, and will do so at a pretty fast pace, so everyone should order a copy ASAP so that they can have it by the end of the first week of the course. This book is amazing and one of those ones you'll want to keep forever and read to your kids.

Schedule

The schedule for this class is somewhat flexible and constantly evolving. The first few weeks, however, have been mapped out pretty tightly.

Wk#	Day	Topic	Homework
1	1/8	Course Intro	Buy text, Socialize, Syllabize, Commando
	1/10	The Command Line #1	See 1/8
	1/12	The Command Line #2	Read Code Ch 1~9
2	1/15	**MLK Holiday**	Reading
	1/17	Binary Numbers	Ch. 10~11
	1/19	Logic Gates	Ch. 12~14, Boolean Logic Lab
3	1/22	Adder/Subractor, Clock	Ch. 15~16
	1/24	Hexadecimal & Memory	Ch. 17
	1/26	The Computer	Ch. 18~20
4	1/29	Computer History & ASCII	Ch. 21~22
	1/31	The Bus & OS	Ch. 23~25
	2/2	Fixed vs. Floating Point Numbers	Finished Code–no HW
5	2/5	Dev Environment	Subdue Sublime

	2/7	Version Control	Git & GitHub
	2/9	Fundamentals Intro	Finish Sublime & GitHub labs, "Hello, World!"
6	2/12	Fundamentals: Symbols	Symbolizer
	2/14	Fundamentals: Operators	Operator
	2/16	Fundamentals: Collections	Collector
7	2/19	Fundamentals: Flow Control	Flow Control
	2/21	Fundamentals: Functions	Functor
	2/23	Fundamentals: Classes	Classy
8	2/26	Objects & Classes	Finish Fundamentals
	2/28	Software Development Process	TBD
	3/2	Picking a Project	Project Problem Statement
SPRING BREAK!!!			
9	3/12	Guest Lecture	TBD
	3/14	Project Idea Presentations	TBD
	3/16	Project Idea Presentations	TBD
10	3/19	Test-Driven Development	TDD Lab
	3/21	TDD	Continue TDD Lab
	3/23	Intro to Code Review	Code Review
11	3/26	In-Class Code Review	Projects
	3/28	In-Class Code Review	Projects
	3/30	In-Class Code Review	Projects
12	4/2	Continuous Integration	Travis-CI Lab
	4/4	Travis-CI	Continue Travis-CI Lab

	4/6	Code Coverage	Codecov
13	4/9	Projects	Projects
	4/11	Projects	Projects
	4/13	Projects	Projects
14	4/16	Practice Project Presentations	Projects
	4/18	Practice Project Presentations	Projects
	4/20	ISAT Senior Symposium	Projects
15	4/23	Project Presentations	None
	4/25	Project Presentations	None
	4/27	First Day of Finals	
End of Semester Exit Interviews			

Important Dates and Deadlines

At the behest of the registrar, a list of dates you may wish to take note of:

- Tuesday, January 16th: Last day of add/drop
- Thursday, January 25th: Last day to add a class with Department Head signature
- Friday, January 26th: Last day to withdraw from JMU with charges canceled

So if I scare you off, get out early. Or conversely, if I turn you on, join us soon!
My academic integrity policy is different from JMU's standard policy, but I will adhere to JMU's standard policies listed on the JMU Syllabus Information for Students page with respect to add/drop, disability accommodations, inclement weather and religious accommodations.

Personal Integrity

First:

> **If I catch you cheating, or doing anything else dishonest, you will fail the course. Period.**

Second, that being said, I strongly encourage sharing and collaboration in most every aspect of the course. That means that I think it's a smart idea for you to:

- Download code you find on the web (include the URL of where you found it and some notes about how you got there)
- Download your classmates' code and use it, even before an assignment is due
- Pay someone to help you write code
- Get code from upperclassmen or people in previous semesters
- Ask your neighbor to give you a hint on a question on a test that you're stumped on
- Use whatever notes, websites, books, or other materials you need to complete most any assignment or test

You'll note that many of the above behaviors would be considered "cheating" in many or most other courses. Here are some guidelines I'd like you to follow:

- **Never EVER copy without attribution**
  Even on tests, if someone or something helped you out, acknowledge it. Make notes in your code if you got it from someone or somewhere else. Copying without attribution is stealing and is a breach of integrity. If you got the code off of the web, there should be a URL and some notes about how you found it. If you paid someone to help you write it, say so.
- **Never copy without understanding**
  The point of the class is to learn and understand stuff. Since you don't get any grades on individual tests or assignments, it's pretty stupid to copy something that you don't understand. Think about it. What point could it possibly serve?
- **Be very hesitant to copy an ENTIRE project**
  While there's a lot to be gained by incorporating parts of your classmates' code in your own project, copying someone else's entire project doesn't really provide you much of a learning experience and wastes people's time.
- **Try to figure it out yourself first**
  90% of writing programs is learning how to write them, and this will stay the same throughout your entire programming career. Being a self-sufficent learner is one of the primary goals of the course.
  Code re-use is a HUGE part of hacker culture. What hackers hate more than anything is not understanding stuff. I want you to get a sense for what it's like to be a part of the fun world of professional hackers.

Okay, so what do I consider a breach of integrity worthy of failure?

- Lying about anything to anyone in the class
- It could be as trivial as the reason why you didn't show up for class or do your part of a group assignment. Everybody screws up sometimes. Don't compound the mistake by

lying about it. We can forgive mistakes but it's VERY difficult to regain trust once it's broken. Swallow your embarrassment or fear and fess up.

- Stealing anything–this includes copying without attribution
- Stealing is just wrong, and since you have a blanket license to copy most any code you can find, there's no reason not to give people credit for the work they did. Passing someone else's original work off as your own is frankly disgusting.
- Threatening, antagonizing, or intimidating anyone in our learning community
- This is unacceptable behavior and will get you at least fired, if not sued in most every company you'd ever work for.

If you are in doubt about something, please ask your prof. Please feel free to come speak to your prof in confidence about anything in this course that troubles you. So far at JMU I've never had a problem with anyone's integrity (that I know about). Don't be the first group to ruin my perfect record. Thanks!

The Lead Coder

Lead Coder–Morgan Benton

Hi, I'm Morgan. I'll be the lead coder for this journey.

As I've said:

Learning is essentially an artifact of relationship

So, in order to lead by example, here is a brief overview of my relationships, who I know and who knows me, and what roles we've played in each other's lives. But first, here's how you can begin relating to me:

Key Info

Office	EnGeo 2118 (office hours/coffee by appointment)
Email	bentonmc@jmu.edu
Cell	973-495-7736
Social	

Academic Community

Undergrad

I got a BA in Leadership Studies and Sociology (with a minor in Physics) in 1996 from the University of Richmond and the Jepson School of Leadership Studies. (Jepson is kinda like ISAT, but for social science.) For my capstone I went to Capitol Hill and surveyed the IT directors of US Senators to try to figure out if the people who were making and passing laws about technology had actually ever used a computer. I was inspired to do this by an online petition I received related to some of the scarier provisions in the Telecommunications Act of 1996. I was supported in my efforts by my professor Joanne Ciulla, whose unorthodox ideas (see her book The Working Life: The Promise and Betrayal of Modern Work) helped me become an outside-the-box thinker. My capstone partner, Elizabeth MacKenzie Biedell, went on to work for the US State Department where she staffed Colin Powell, the CIA where she briefed the White House, and has become an advocate for information sharing and against the over-classification of intelligence.

As an undergrad I did a lot of community service. I helped Richmond Habitat for Humanity raise close to $50,000 and build four houses. I got schooled in basketball by kids at the Sacred Heart Center while I was volunteering in their after school program. I helped feed meals to homeless men, and learned a lot from them in the process. I left college with a deep commitment to serve, inspired by the words of one of my best friends, the Associate Chaplain David Dorsey, who said:

There's nothing very "high" about "higher education" if it doesn't give something back to the community from which it was taken.

Another of my mentors was Richard A. Couto (who insisted that we call him Dick–very hard to do with a straight face if you're my age). I helped Dr. Couto produce An American Challenge: A Report on Economic Trends and Social Issues in Appalachia. My job was to turn US census data into maps–perhaps the first time that I employed computer technology to help effect social change, this time for poor people in the Appalachian region.

These are just a few of the many people with whom I connected as an undergrad, who influenced me, and with whom I continue to connect to solve real problems for real people.

Grad School

I got an MS and PhD in Information Systems from New Jersey Institute of Technology. It took me TEN YEARS (1998-2008)! For my dissertation, I built a web-based system to help teachers write better multiple-choice questions (MCQs). (Spoiler: it didn't really work.) In actuality, I hate multiple-choice questions and multiple-choice tests. My system was designed as a Trojan horse (figuratively–it was NOT spyware) to convince teachers that MCQs are more trouble than they are worth. This grueling journey solidified my lifelong ambition to radically change education globally.

Along the way, I was lucky to receive guidance from Dylan Wiliam, one of the world's foremost scholars on formative assessment. I met Dr. Wiliam while I was a graduate intern at ETS (you know, where they make the SAT and GRE), and he taught me a painful and enduring lesson–technology is not always the answer. His 1998 meta-analysis of hundreds of studies of classroom assessment has grounded and guided my teaching practice and research ever since.

Starr Roxanne Hiltz and Murray Turoff were the Distinguished Professors at NJIT who gave me my job there and made it possible for me to support my family while getting my PhD. They predicted the age of the Internet way back in 1978 in their book The Network Nation. They were pioneers in online education, and there are still many of their ideas that haven't yet been implemented in modern online learning management systems.

I was teaching four courses per semester throughout my time at NJIT. I may have learned more from my students than from any other source. It was from my students that I learned humility, flexibility, and to see just how vicious the system is that binds us all together. I still have relationships with a number of them, and they continue to keep me on my toes.

Of course, my most important relationship in grad school was with my advisor, Marilyn Tremaine. Marilyn was one of the founding mothers of the field of human-computer interaction (HCI) and helped create one of the most vibrant and successful academic communities (the ACM's SIGCHI) of the last several decades. She let me live in her house. She showed me that even if I've been a total slacker, there's always something productive we can do with this week's meeting.

JMU

I began at JMU in August 2006 and have been here for over a decade. It is here that I've had the opportunity and freedom to put into practice many of the innovations in teaching that I've discovered over the years. The ISAT department deeply values creativity and autonomy, so while my colleagues may have offered (sometimes sharp) criticism over the years at my methods, nobody ever told me to stop, and many encouraged me to keep moving forward. I deeply value this aspect of the ISAT Department–the willingness to take risks to achieve great things.

That being said, it is my students who have had the biggest influence on me. It was Brian Rapp (ISAT '10), who, as my TA, first goaded me into trying my choose-your-own-grade strategy back in the spring of 2009. It was my capstone students who helped me write much of the software that I use in my teaching. It continues to be students like Cyril Thornton ('10), Adam Maas ('12), Chiedo John ('13), and Josh Erney ('14) who regularly ping me about new tech developments that help keep me at the cutting edge of my field. I feel deep gratitude every time a student opens up to me and allows me to play a part in their story.

I would be remiss if I didn't mention Nicole Radziwill. We teach together, research together, play together, live together. We started the Burning Mind Project together. We've published over half a dozen papers together and applied for over a million dollars in funding together. We've been to Burning Man three times and it is from her that I am constantly challenged to try new things, examine who I am, and continually reinvent myself.

There's plenty more to learn about me, and I strongly encourage everyone to invite me out for coffee or lunch or something! I look forward to getting to know each and every one of you better.