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

Spring 2017

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, your semester grade will drop by one letter. Woody Allen said, “80% of success is just showing up.” I care if you’re here.

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	1/9	Course Intro	Buy text, Socialize, Syllabize, Commando
	1/11	The Command Line #1	See 1/9
	1/13	The Command Line #2	Read Code Ch 1~9
2	1/16	MLK Holiday	Reading
	1/18	Binary Numbers	Ch. 10~11
	1/20	Logic Gates	Ch. 12~14, Boolean Logic Lab
3	1/23	Adder/Subractor, Clock	Ch. 15~16
	1/25	Hexadecimal & Memory	Ch. 17
	1/27	The Computer	Ch. 18~20
4	1/30	Computer History & ASCII	Ch. 21~22
	2/1	The Bus & OS	Ch. 23~25
	2/3	Fixed vs. Floating Point Numbers	Finished Code–no HW
5	2/6	Dev Environment	Subdue Sublime
	2/8	Version Control	Git & GitHub
	2/10	Fundamentals Intro	Finish Sublime & GitHub labs, "Hello, World!"
6	2/13	Fundamentals: Symbols	Symbolizer
	2/15	Fundamentals: Operators	Operator
	2/17	Fundamentals: Collections	Collector
7	2/20	Fundamentals: Flow Control	Flow Control

	2/22	Fundamentals: Functions	Functor
	2/24	Fundamentals: Classes	Classy
8	2/27	Objects & Classes	Finish Fundamentals
	3/1	Software Development Process	TBD
	3/3	Picking a Project	Project Problem Statement
SPRING BREAK!!!			
9	3/13	Guest Lecture	TBD
	3/15	Project Idea Presentations	TBD
	3/17	Project Idea Presentations	TBD
10	3/20	Test-Driven Development	TDD Lab
	3/22	TDD	Continue TDD Lab
	3/24	Intro to Code Review	Code Review
11	3/27	In-Class Code Review	Projects
	3/29	In-Class Code Review	Projects
	3/31	In-Class Code Review	Projects
12	4/3	Continuous Integration	Travis-CI Lab
	4/5	Travis-CI	Continue Travis-CI Lab
	4/7	Code Coverage	Codecov
13	4/10	Projects	Projects
	4/12	Projects	Projects
	4/14	Projects	Projects
14	4/17	Practice Project Presentations	Projects
	4/19	Practice Project Presentations	Projects

	4/21	ISAT Senior Symposium	Projects
15	4/24	Project Presentations	None
	4/26	Project Presentations	None
	4/28	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 17th: Last day of add/drop
- Thursday, January 26th: Last day to add a class with Department Head signature
- Friday, January 27th: 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 Prof

My research mainly involves coming up with pedagogical alternatives that maximize student motivation and learning. Being a tech geek, web-based technology plays a pretty heavy role in what I came up with.

My favorite part of my job is getting to hang out with students and play with technology. Feel free to come see me any time. My info:

Office	ISAT/CS 124
Office Hours	Make an appointment
Mobile	973-495-7736 (calls and texts are ok within reason)
Email	bentonmc@jmu.edu
Facebook	http://www.facebook.com/morgan.benton
Twitter	http://twitter.com/morphatic
Blog	http://www.burningmindproject.org