****

 **CSIS-110**

**Introduction to Computer Science**

**Spring 2018**

**Study Tips for Final Exam**

The final exam will be given on **Friday, May 4th, from 6:30 – 8:30 pm** **in RB202**. As with the previous exams, you may bring one 8 1/2 by 11 inch sheet of paper with anything you want, using both sides. Your sheet must be submitted when you hand in your exam, it must have your name on it, and it must be your own work. If your sheet does not meet the requirements, you will not be allowed to use it or you may lose points from your exam. **Calculators and cell phones are not permitted.**

The exam is cumulative. **Approximately 45% of the exam will be on programming.** The exam will cover all the assigned readings from the textbooks, in addition to the material covered in class, on the homework, and in lab. Reviewing labs, homework, and previous exams is a great way to prepare for the final exam. Study hard and good luck!

**Office hours for exam week:** **Matthews**: Tuesday 2-4, Wednesday 2-4, Thursday 2-4, Friday 2-4

 **Egan**: Tuesday 2-4, Wednesday 1-3, Thursday 11-1, Friday 11-1

**Review session (courtesy of Nick Murabito):** Wednesday 9-11pm in RB328

Potential Exam Topics (not necessarily complete):

# Data Representation

* Binary numbers
* ASCII, Unicode
* Sign magnitude representation
* Base 2 to base 10 conversions
* Base 10 to base 2 conversions
* Octal, hexadecimal
* Number of things that can be represented with n bits
* Overflow
* Text compression: Huffman, run-length
* Compression ratio
* Representing color (blue, green, red components)
* Representing images (digitization of images, pixels)
* Representing audio (sampling, mp3 format)

# Boolean Expressions, Gates, Circuits

* Boolean expressions (AND, OR, NOT operations)
* AND, OR, and NOT notations
* Truth tables
* Gates (AND, OR, NOT)
* Circuits
* Sums of products circuit construction algorithm

# Computing Components

* von Neumann Architecture
* Memory
* Memory addresses
* CPU
* Control unit
* Arithmetic logic unit (ALU)
* Fetch - decode - execute cycle
* Kilobyte, megabyte, gigabyte, terabyte, …
* Disk seek, latency, and transfer time, in general
* Disk geometry - sectors, tracks, surfaces, platters
* Primary (RAM) vs. secondary memory

# Machine Language

* Machine instructions such as Load, Store,

Compare, Jump, Add, Sub, Decrement, Increment

* Opcode and address
* Translation of Python code into assembly language
* Translation of assembly language to Python code

# Operating Systems

* Responsibilities of the operating system
* Memory management approaches using dynamic partitions:
	+ First fit
	+ Best fit
	+ Worst fit
* Multiprogramming
* Process management approaches:
	+ First come first serve
	+ Shortest job next
	+ Round robin
	+ Average turnaround time for each
	+ Gantt charts

# Artificial Intelligence

* Understand types of tasks computers do better than humans and types of tasks humans do better than computers
* Turing test and chatbots
* Eliza and her rules
* Strong vs. weak equivalence
* Artificial neurons
* Games

### Et Cetera

* basic questions on Alice
* basic questions on Scheme

#

# Python (for Python Multimedia sections)

* The JES Python environment (command area vs. program area, etc.)
* Naming things
* Basic data manipulation (print, input, arithmetic, …)
* Types of values (integer, string, real number, picture, etc.)
* Names, values, and function results are interchangeable
* Defining a function
* “for” loop to modify all pixels in a picture
* Accessing individual pixels
* Changing pixel colors (negating, increasing red, grayscale, etc.)
	+ getRed, getBlue, getGreen
	+ setRed, setBlue, setGreen
	+ makeColor, getColor, setColor
* Nested “for” loops to process a range of pixels in a picture
* Mirroring algorithms
* Copying a range of pixels (or whole picture) into another picture
* Growing/shrinking a picture
* Color distance
* “if” statements
* Representing sounds
* Altering sound sample values
* “for” loop to modify all samples in a sound
* “for” loop to modify a range of samples in a sound
* Functions:
	+ Parameters
	+ Changing values of parameters
	+ Returning values
	+ Calling/testing
* Clipping and copying sounds
* Altering sound volume
* getSampleValue, getSampleValueAt, etc.
* Altering sound frequency
* Reversing sounds
* Splicing sounds to create new sounds
* Drawing objects (ovals, text, etc.)
* Animations
* There might be some short-answer questions on Python.
* There will definitely be Python questions in which we give you some code and ask you what it does or to fix it.
* There will also be Python questions where we ask you to write code to solve a particular problem.