

CSE 1341 Principles of Computer Science I

ASIM 1310 Art & Code

May Term: 2016

Instructor: Ira Greenberg

Teaching Assistant: Zizhen Chen

email: igreenberg@smu.edu

Office: 214-768-4625

Lecture 1 – 5 PM

Labs: t.b.d.

Classroom: Center of Creative Computation, Owen)

Office Hours: Owen: Daily 12 – 1PM and by appointment

<http://iragreenberg.com>

<http://iragreenberg.net>

follow me: @processingTips, @iragreenberg

Course Description:

This course presents an introduction to computer science, programming and object-oriented design from a “Creative Coding” perspective. Course examples, programming exercises, assignments and exams will incorporate computer graphics and interactivity, using both the Processing and Java programming languages. Object-Oriented Programming (OOP) principals will be covered, including encapsulation, composition, inheritance and polymorphism.

Required Text:

Processing: Creative Coding and Generative Art in Processing 2, Ira Greenberg, Dianna Xu, Deepak Kumar, 2013, friends of ED

Recommended Texts:

- *Processing: Creative Coding and Generative Art in Processing 2*, Ira Greenberg, Dianna Xu, Deepak Kumar, 2013, friends of ED
- *Processing: Creative Coding and Computational Art*, Ira Greenberg, 2007, friends of ED
- *Processing: A Programming Handbook for Visual Designers and Artists*, Casey Reas, Ben Fry, 2007, MIT Press
- *Learning Processing*, Daniel Shiffman, 2008, Morgan Kaufmann
- *Visualizing Data*, Ben Fry, 2007, O'Reilly Press
- *Deitel & Deitel: Java How to Program: Late Objects Version*, 8/E, Pearson Prentice Hall

Links:

[Processing Related and Code Art](http://processing.org)

<http://processing.org>

<http://openprocessing.org>
<http://reas.com>
<http://benfry.com>
<http://www.shiffman.net>
<http://cs.nyu.edu/~perlin/>
<http://roberthodgin.com>
<http://infosthetics.com>
<http://www.smashingmagazine.com/2007/08/02/data-visualization-modern-approaches>
<http://toxi.co.uk>
<http://www.tobyjoe.com/2009/07/toward-the-bare-metal>
<http://horizon.wiki.nmc.org/2010+Data+Visualization>
<http://design.osu.edu/carlson/history/lesson9.html>

Java

<http://download.oracle.com/javase/6/docs/api/>
<http://download.oracle.com/javase/tutorial/>
<http://www.java.com/en/>
<http://www.oracle.com/technetwork/topics/newtojava/overview/index.html>
<http://www.oracle.com/technetwork/java/javase/overview/index.html?ssSourceSiteId=ocomen>
<http://www.oracle.com/us/technologies/java/index.html>
<http://math.hws.edu/javanotes/>

Course Requirements:

This course will include readings, programming assignments, quizzes, presentations and critiques. The due date for all assigned materials will be announced in advance. It is the student's responsibility to have all assignments ready on time.

Any student who has to be absent on an assignment due date must arrange to have the assignment submitted early. In addition, it is the student's responsibility to make up any missed work or locate Topic notes due to absence.

Method of Presentation:

Limited Topics will introduce the Processing programming language along with project concepts and specifications. This course is NOT designed as a traditional "sage on the stage" Topic. Class time will be used to discuss concepts and project issues, work collaboratively and to ask questions. Students need to come to class prepared to use this time effectively. Being unprepared is equivalent to an absence.

Method of Evaluation:

There will be programming assignments, quizzes, exams and critiques. Attendance and proactive participation in class and lab is also expected. Points will be awarded as described below.

Total: 100 points

Programming Assignments = 30 points

Quizzes = 30 points

Exams = 30 points

Class/Lab Attendance and Participation = 10 points

Quizzes that are missed cannot be made up. The lowest quiz grade of the semester will be dropped

Final grades are determined as follows:

- * 95 - 100 : A
- * 90 - 94 : A-
- * 86 - 89 : B+
- * 83 - 85 : B
- * 80 - 82 : B-
- * 76 - 79 : C+
- * 73 - 75 : C
- * 70 - 72 : C-
- * 66 - 69 : D+
- * 60 - 65 : D
- * 0 - 60 : F

Learning Objectives:

After successful completion of this course, you should be able to:

1.0 – DEMONSTRATE COMPETENCY IN PROGRAM CONCEPTUALIZATION AND DESIGN

- 1.1 Follow a detailed process for analyzing a program's requirements
- 1.2 Use UML or other graphical process to design a solution to a problem.
- 1.3 Design, code and document a programming project.

2.0 – DEMONSTRATE COMPETENCY IN ALGORITHMIC DEVELOPMENT

- 2.1 Understand memory management and dynamic allocation.
- 2.2 Develop a complex and efficient algorithm
- 2.3 Solve a problem iteratively and recursively\
- 2.4 Use single and multi-dimensional arrays in a programming project.
- 2.5 Implement a variety of File I/O techniques

3.0 – DEMONSTRATE COMPETENCY IN FUNDAMENTAL PROGRAMMING THEORY

- 3.1 Understand primitive and reference variables.
- 3.2 Create parameterized functions.
- 3.3 Locate and explain syntax errors in a program.
- 3.4 Use techniques for debugging programs.
- 3.5 Understand how to compile and execute a program.

4.0 – DEMONSTRATE COMPETENCY IN OBJECT-ORIENTED PROGRAMMING

- 4.1 Create programmer-defined objects
- 4.2 Use existing class libraries to develop collections of objects
- 4.3 Use composition in a programming project.
- 4.4 Use inheritance in a programming project.
- 4.5 Explain the ideas behind polymorphism.
- 4.6 Explain the ideas behind object reusability and modularity.

Programming Assignments and Labs

Each day you will be assigned a programming challenge which is due by 11am the following day.

The lab assignments must be submitted to Blackboard before the due date. If submitted late you will lose as much as 30% of your grade for the assignment.

Make sure that you can access Blackboard (<http://courses.smu.edu>) and that you are enrolled correctly in your course! Email your instructor immediately if you are not correctly enrolled.

Please email help@lyle.smu.edu with any login problems you are having with Windows or Unix. Login problems with Blackboard should be directed to your instructor. Include your 8-digit SMU id in your email. **All login problems must be resolved in the first day of class.**

Attendance is taken daily. You will **not be counted** if you come late or leave early unless you can demonstrate that you have completed that week's programming assignment correctly and understand the topic.

Daily Schedule of Course Topics

(Please do not work ahead, as topics presented may change. If you seek greater challenges, please come speak with me, and I will most happily accommodate you.)

Course Topics:

Topic 1:

- Admin/policy
- History of computing
- Computation and art
- Left | right brain integration

Topic 2:

- Languages and hardware
- Java and Processing
- Processing Environment/coordinate system

- Topic 3:
- Naming
- Statements
- Expressions

Example: Hello World Enhanced (multiple but discrete ellipses)

Topic 2:

- Variables (Primitive)
- Assignment
- Order of operations
- Syntax
- State/color (state machine)
- Accumulator model

Assignment: T.B.D

Day 3:

Topic 1:

- Definite Loops (while)
- Conditional logic
- Functions
- Parameters/scope
- Return value
- Modifying parameter values

Example: Fancy Hello World example (3D sphere using overlapping gradients);

Topic 2:

- Arrays
- Points (int, truncation/float)
- Drawing (alpha/transparency, anti-aliasing)
- Lines (line Primitive vs vertex, inaccuracy)
- beginShape, endShape

Assignment: T.B.D

Day 4:

Topic 1:

- Randomization
- Primitives (Ellipse, Rectangle, Triangle, Arc)
- Algorithmic Approach
- Pseudo code

Example: Polygon/Star (sin/cos) (Polar coord, casting/type conversion)

Topic 2:

- Simple 2D transformations
- Matrix stack
- Pushing and popping
- Transformations are cumulative

Assignment: T.B.D

Day 5:

Topic 1:

- Curves (polynomials)
- Bezier

Example: curve Mandala (or Flower, Spiral)

Topic 2:

- Spline
- curveVertex

Assignment: T.B.D

Day 6:

Topic 1:

Midterm Review

Topic 2:

Midterm Exam

Day 7

Topic 1:

- Classes
- Data members, methods
- Objects, instantiation and encapsulation
- Abstraction and modules

Example: Polygon class

Topic 2:

- Class design
- Polymorphism
- Inheritance
- Interfaces
- Patterns

Assignment: T.B.D

Day 8:

Topic 1:

- Color theory

- Number systems
- Packed 32-bit integer twos complement system: bit-wise ops
- Models (RGB/HSB)

Example: Color gradients (linear/radial/wave)

Topic 2:

- Anatomy of an image (pixels)
- PImage type
- Pixel Buffer (per Window and per Image)
- Gradients
- Processing API blends/filters

Assignment: T.B.D

Day 9:

Topic 1:

- Frame vs time based animation (Processing's draw function)
- FrameRate (move something across screen -- simple accumulator model)
- Fade
- Speed/direction

Example: Animate Polygon

Topic 2:

- Wall collision
- Acceleration, Damping, friction
- Simulating physics (Newtonian Integration)

Assignment: T.B.D

Day 10:

Topic 1:

- Interactivity and Events
- Object-object collision

Example: Pong

Topic 2:

- Game Programming
- atan()
- ArrayList
- Score/text

Assignment: T.B.D

Day 11:

Topics 1 & 2:

- 3D

Assignment: Arcade Game Continued

Attendance Policy

Students are expected to attend all classes. If a student is absent from class on the due date of any assignment, they are expected to make alternative arrangements to assure that the assignment is turned in on time.

Academic Honesty and Misconduct - The Honor Code

All Code you create in this course MUST be your own, or clearly stated otherwise—NO EXCEPTIONS.

All work undertaken and submitted in the course is governed by the University's Honor Code. The relevant section of the Code, taken from the Preamble of the Honor Council's Constitution:

Intellectual integrity and academic honesty are fundamental to the processes of learning and of evaluating academic performance, and maintaining them is the responsibility of all members of an educational institution. High personal standards of honesty and integrity are a goal of education in all the disciplines of the University. Students must share the responsibility for creating and maintaining an atmosphere of honesty and integrity. Students should be aware that personal experience in completing assigned work is essential to learning. Permitting others to prepare their work, using published or unpublished summaries as a substitute for studying required materials, or giving or receiving unauthorized assistance in the preparation of work to be submitted are directly contrary to the honest process of learning. Students who are aware that others in a course are cheating or otherwise acting dishonestly have the responsibility to inform the professor and/or bring an accusation to the Honor Council.

A violation of the Honor Code may result in an "F" for the course, and the student may be taken before the Honor Council. If you are unclear about this policy, either in general or in its specific application, please see the instructor. The Honor Code is in the SMU Student handbook and may be viewed on-line at :

http://smu.edu/studentlife/PCL_05_HC.asp

*** Disability Accommodations:** Students needing academic accommodations for a disability must first contact Disability Accommodations & Success Strategies (DASS) at 214-768-1470 or www.smu.edu/alec/dass.asp to verify the disability and to establish eligibility for accommodations. They should then schedule an appointment with the professor to make appropriate arrangements. (See University Policy No. 2.4; an attachment describes the DASS procedures and relocated office.)

*** Religious Observance:** Religiously observant students wishing to be absent on holidays that require missing class should notify their professors in writing at the

beginning of the semester, and should discuss with them, in advance, acceptable ways of making up any work missed because of the absence. (See University Policy No. 1.9.)

*** Excused Absences for University Extracurricular Activities:** Students participating in an officially sanctioned, scheduled University extracurricular activity should be given the opportunity to make up class assignments or other graded assignments missed as a result of their participation. It is the responsibility of the student to make arrangements with the instructor prior to any missed scheduled examination or other missed assignment for making up the work. (University Undergraduate Catalogue)