CSCI 144 – Spring 2015 Handout: Syllabus
Intro to CS: Multimedia
Michael H. Goldwasser
Saint Louis University Tuesday, 13 January 2015

CSCI 144
Introduction to Computer Science: Multimedia

Contents

1 Overview **1**
1.1 Catalog Description 1
1.2 Detailed Description 2

2 Course Administration **2**
2.1 The Staff 2
2.2 Class Meetings 2
2.3 Textbook 3

3 Online Resources **3**
3.1 CSCI 144 Web Page: `http://cs.slu.edu/~goldwasser/144/` 3
3.2 Email with Instructor 3
3.3 Electronic Assignment Submission 3

4 Course Grades **4**
4.1 Graded Work 4
4.2 Course Grades 4
4.3 Late Policy 5
4.4 Academic Integrity 5

5 Additional Information **6**
5.1 Supporting Student Success 6
5.2 Tutoring Resources at SLU 7
5.3 Computing Resources at SLU 7

1 Overview

1.1 Catalog Description

"An introduction to computer programming motivated by the creation and manipulation of images, animations, and audio. Traditional software development concepts, such as data representation and control flow, are introduced for the purpose of image processing, data visualization, and the synthesis and editing of audio."

1.2 Detailed Description

As an introduction to the field of computer science, we will explore an interesting dichotomy of precision and creativity, which feeds the development of new tools, algorithms, and data analyses. The precision and attention to detail is required because as "powerful" as computing technology seems, the machines simply follow the instructions they are given, and if they are given flawed instructions, then they will not behave as desired. However, computer science goes well beyond mere coding. Creativity allows us to envision new uses for computing, and to develop new tools or find new designs that improve on existing technologies.

In this course, we further emphasize the creative process of software design by focusing on the creation of static, animated, and interactive visualizations, as well as the manipulation of existing imagery and audio. We will use the Processing programming language (more specifically, Processing 2). Processing is an open-source programming language that was originally developed in the MIT Media Lab as a crossover between graphic design and computer programming. Additional references about the language can be found in the "documentation" section of the course web page.

2 Course Administration

2.1 The Staff

Instructor:	Dr. Michael Goldwasser
Email:	`goldwamh@slu.edu`
Web:	`http://cs.slu.edu/~goldwasser`
Office:	Ritter Hall 108
Telephone:	(314) 977-7039
Office hours:	Tuesdays 10:00–10:50am
	Wednesdays 2:00–3:00pm
	Thursdays 10:00–10:50am
	or by appointment

2.2 Class Meetings

The Lectures

The material will be presented in two weekly meetings, which will be a combination of lecture and studio style. Attendance is expected and class participation is most welcome. These meetings will offer learning opportunities that cannot be recreated purely from readings. Information on the daily topics can be found on the course schedule web page.

Time:	Tue/Thu, 11:00am–12:15pm
Place:	Ritter Hall 316

2.3 Textbook

The required textbook for this course is:

Title:	*Processing: Creative Coding and Generative Art in Processing 2*
Authors:	Ira Greenberg, Dianna Xu, and Deepak Kumar
Publisher:	friends of ED, APress, 2013
ISBN:	978-1-4302-4464-6
Website:	`www.apress.com/9781430244646`

The text should be available through the campus bookstore as well as various online book vendors. We have also place a copy of it on reserve in Pius library.

3 Online Resources

3.1 CSCI 144 Web Page: `http://cs.slu.edu/~goldwasser/144/`

With the exception of the first day's printed handouts, most of the information for this course will be distributed only by means of the course web page. This web site will contain all assignments, a schedule of lectures, examples from lecture, and links to many other sources of information.

The web page contains some information (e.g. solutions, submitted assignments, individual grades) that is more sensitive and therefore which will be available to students in the class only after they have identified themselves properly. To gain access to these parts of the web page, a student must first complete an online questionnaire, creating a unique identity and password.

3.2 Email with Instructor

Face-to-face contact in class and in office hours is most desirable. Yet email is a convenient form of communication as well. I try to respond to email promptly, including at least once each evening when possible.

3.3 Electronic Assignment Submission

Programming projects for this course must be submitted electronically! The submission procedure will be done through the course web page, and allows students to submit from any computer connected to the Internet. Each student in this class will be selecting a unique username/password combination solely for use in identifying the student when using the course web page. Details of the procedure are discussed at:
`http://cs.slu.edu/~goldwasser/144/submit/`

4 Course Grades

4.1 Graded Work

Course grades will be based on the following components:

- **Projects (49%)**

  We expect there to be a total of 7 programming projects during the course, each worth 7% of the final grade. The assignments will be submitted electronically and generally due at 11:59pm on the assigned due date (not that you need to be working on it until that time).

- **Homework Assignments (18%)**

  We expect there to be a total of 6 written homework assignments during the course, each worth 3% of the final grade. These will generally be due at the beginning of class on the assigned due date. The problems will consist of short questions reinforcing technical material in the lectures and readings (and questions that are more akin to what might be found on exams).

- **Exams (33%)**

  - **Midterm Exam (13%)**, Thursday, 5 March 2015, 11:00am–12:15pm
  - **Final Exam (20%)**, Tuesday, 12 May 2015, 12:00-1:50pm

4.2 Course Grades

Letter grades will be based on each students overall percentage of awarded points according to the following formula.

Student percentage above 90% will result in a grade of A or better.
Student percentage above 87% will result in a grade of A- or better.
Student percentage above 83% will result in a grade of B+ or better.
Student percentage above 80% will result in a grade of B or better.
Student percentage above 77% will result in a grade of B- or better.
Student percentage above 73% will result in a grade of C+ or better.
Student percentage above 70% will result in a grade of C or better.
Student percentage above 67% will result in a grade of C- or better.
Student percentage above 60% will result in a grade of D or better.
Student percentage below 60% will result in a grade of F.

Any modification to this scale at the end of the year will be *in favor of the students*. That is we may later decide to award an A to a student who is slightly below the above cutoff, but we certainly will not deny an A from someone who is above the cutoff.

4.3 Late Policy

All exams must be taken promptly at the required time. Requests for rescheduling an exam will only be considered if the request is made prior to the start of the exam, or else in an "emergency" situation with appropriate documentation.

For both written assignments and programming projects, we wish to allow students to continue to work comfortably beyond the official deadline when a little more time will result in more progress, while at the same time discourage students from falling significantly behind pace and jeopardizing their success on future assignments. Our solution is the following exponentially decaying late formula.

We will consider an assignment submission "complete" when any part of the assignment is last submitted or modified. Any assignment that is not complete promptly by its due date and time will be assessed a penalty based on the formula $S = R \cdot e^{-h/173}$, where S is the grade given, R is the grade the work would have received had it been turned in on time, and h is the amount of time (in hours or fractions thereof) that the work was late. Examples:

— work turned in 1 hour late receives over 99.6% of its original credit
— work turned in 5 hours late receives over 97% credit
— work turned in one full day late receives less than 88%
— work turned in two full days late receives less than 76%
— work turned in five days late receives less than 50%

The above policies will be waived only in an "emergency" situation with appropriate documentation.

4.4 Academic Integrity

Students are expected to have read and abide by the University statement on Academic Integrity available on page 2 of the Policies and Procedures section of the Saint Louis University's Undergraduate Catalog. A more detailed policy statement is given by the College of Arts & Science (`www.slu.edu/colleges/AS/academic_honesty.html`), also applying to this course.

In addition to those general statements, we wish to discuss our policy in the context of this course. When it comes to learning and understanding the **general course material**, you may certainly use other reference materials and you may have discussions with other students in this class or other people from outside of this class. This openness pertains to material from the text, general syntax and use of the Processing language, and other computing tools. At various times, we may look to the website `www.openprocessing.org` for inspiration, yet that must be limited to solely inspiration (not source code).

When it comes to **work that is submitted for a grade in this course**, you are not to search for or use any direct or indirect assistance from unauthorized sources, including but not limited to:

- other students in this class

- past students, whether from this school or other schools
- other acquaintances
- other texts or books
- Processing projects found online

Acceptable sources of information include consultations with the instructor, teaching assistants, or members of organized tutoring centers on campus, as well as any materials explicitly authorized in an assignment. Even in these cases, if you receive significant help you should make sure to document both the source of the help as well as the extent.

On certain programming assignments, we may explicitly allow students to work in pairs. In this case, conversations between partners is both permissible and required. Furthermore, both students are expected to contribute significantly to the development of the submitted work. It is unethical to allow a partner to "sign on" to a submission if that partner did not significantly contribute to the work.

Any violations of these policies will be dealt with seriously. Penalties will apply as well to a student who is aiding another student. Any such violations will result in a minimum penalty of a zero on the given assignment that cannot be dropped, and severe or repeated violations will result in an immediate failing grade in the course. Furthermore all incidents will be reported in writing to the Department and/or the Dean, as per the College procedure.

5 Additional Information

5.1 Supporting Student Success

In recognition that people learn in a variety of ways and that learning is influenced by multiple factors (e.g., prior experience, study skills, learning disability), resources to support student success are available on campus. Students who think they might benefit from these resources can find out more about:

- Course-level support in the remainder of this section or by asking the instructor
- University-level support (e.g., tutoring services, university writing services, disability services, academic coaching, career services, and/or facets of curriculum planning) by visiting the Student Success Center (BSC 331) or by going to `www.slu.edu/success`.

Students with a documented disability who wish to request academic accommodations are encouraged to contact Disability Services to discuss accommodation requests and eligibility requirements. Please contact Disability Services, located with the Student Success Center, at `disability_services@slu.edu` or 314-977-3484 to schedule an appointment. Confidentiality will be observed in all inquiries. Once approved, information about academic accommodations will be shared with course instructors via email from Disability Services and viewed within banner via the instructor's course roster.

5.2 Tutoring Resources at SLU

Our department employees many junior/senior computer science majors to help out with students in introductory courses. They might not be familiar with the Processing language, but they should be eager to help.

Our department web page maintains a current list of the available times and locations at `cs.slu.edu/undergrad-cs/lab-hours`.

As stated in the Academic Integrity policy of Section 4.4, these tutors are an acceptable resource for help, yet you should still document both the source of the help as well as the extent, if significant.

5.3 Computing Resources at SLU

Processing is installed in Ritter 316 and in Ritter 225, however those rooms are not generally open and available outside of classtime. Fortunately, Processing is freely available software for all major computing platforms and can be installed elsewhere.

Our department runs a Linux computer server named `turing` that serves as the primary computing environment for many of our other courses, and a lab in Ritter 121. An account will be created for you to use that system (in person or remotely) and Processing will be installed there as well. See `cs.slu.edu/computing-resources` for further documentation regarding use of turing's facilities.