San José State University
College of Science / Department of Computer Science
CS-160, Software Engineering, Section 3, Spring 2017

Course and Contact Information

Instructor:	Robert Bruce
Office Location:	Duncan Hall, DH-282
Telephone:	(707) 483-9340
Email:	Robert.Bruce@sjsu.edu
Office Hours:	Mondays and Wednesdays 8pm-9pm, or by appointment
Class Days/Time:	Monday and Wednesday, 6:00PM-7:15PM
Classroom:	SCI-311
Prerequisites:	CS-146 and CS-151 (with a grade of "C-" or better in each); CS-100W (with a grade of "C" or better) or instructor consent.

Course Format

Technology Intensive, Hybrid, and Online Courses

This class is technology intensive. You will need an Intel x86 or compatible laptop or desktop machine. We will be working in a Linux development environment. You will be using command line tools such as Make, gcc, etc. You will be creating and managing SQL tables in Postgres. You will also be creating static and dynamic web pages using the Apache web server. You will need an internet connection to access materials on the course website and Canvas. I will be providing a 120GB solid state drive configured with Linux for you with a USB3.0 or two USB2.0 interfaces. Generally, Mac notebooks and Windows-based notebooks work fine. The most difficult task is getting the external Linux drive to boot into secure boot environments. All software and software development tools we will be using are open source. You may use a variety of languages for your programming project including Java, C, C++, PhP, Perl, Python, etc.

Faculty Web Page and MYSJSU Messaging

All course materials (handouts, assignments, lectures, etc.) will be posted to my course web site at: http://www.cs.sjsu.edu/~bruce/. You will be using Canvas to submit your programs and receive feedback/grades on your assignments and exams. You are responsible for regularly checking with the email messaging system through MySJSU at http://my.sjsu.edu to learn of any updates.

Course Description

Software engineering principles, software process and process models, requirements elicitation and analysis, design, configuration management, quality control, project planning, social and ethical issues. Required team-based software development, including written requirements specification and design documentation, oral presentation, and tool use. Prerequisite: CS 146, CS 151 (with a grade of "C-" or better in each); CS 100W (with a grade of "C" or better) or instructor consent. Computer

Science and Software Engineering Majors only.

Course Learning Outcomes (CLO)

Upon successful completion of this course, students will be able to:

1. Course Learning outcome: Design a scalable, relational database schema to support a computer vision face detection pipeline.

2. Course Learning outcome: Apply the Postgres application programming interface in software applications.

3. Course Learning outcome: Use the Active Appearence Modelling (AAM) algorithm from OpenFace to locate facial features.

4. Course Learning outcome: Use Delaunay triangulation from the OpenCV library to create a face mesh.

5. Course Learning outcome: Use the FFMPEG library to convert video into a series of still images.

6. Course Learning outcome: Use the FFMPEG library to extract video metadata information (frames per second, number of frames, etc.).

7. Course Learning outcome: Use named pipes, fork(), and exec() to implement inter-process communication in the face detection pipeline.

8. Course Learning outcome: Create dynamically-driven web page content using Apache.

9. Course Learning outcome: Apply software engineering principles through the entire development lifecycle from design, implementation, testing, release, and maintenance.

Required Texts/Readings

Textbook

Beginning Software Engineering by Rod Stephens. ISBN-13: 978-1118969144.

I've put three personal copies of this book on course reserves for twenty-four hour checkout from the SJSU King Library. Alternately, you can purchase this book from an online bookstore such as Amazon, Barnes and Noble, etc.

Other Readings

PostgreSQL: Up and Running by Regina Obe and Leo Hsu.

This book is available electronically from the SJSU King Library at http://libaccess.sjlibrary.org/login?url=http://proquest.safaribooksonline.com/?uiCode=Incommon-member@co.calstate.edu&xmlId=9781449373184.

This book is also available electronically from San Jose Public Library at http://0-

Learning OpenCV: Computer Vision with the OpenCV Library by Gary Bradski and Adrian Kaehler.

This book is available electronically from the SJSU King Library at http://sjsu.eblib.com/patron/FullRecord.aspx?p=443167.

Advanced programming in the UNIX environment by W. Richard Stevens and Stephen A. Rago.

This book is available electronically from the SJSU King Library at http://libaccess.sjlibrary.org/login?url=http://proquest.safaribooksonline.com/?uiCode=&xmlId=9780321638014.

This book is also available electronically from San Jose Public Library at http://0-proquest.safaribooksonline.com.catalog.sjlibrary.org/?uiCode=califa&xmlId=9780321638014.

Other technology requirements / equipment / material

You will be provided with a 120GB solid state drive with a USB3.1 interface. This drive has been pre-configured with Linux Mint in a graphical X-windows environment along with a host of GNU development and programming utilities, a Postres database server, the OpenCV library, OpenFace library, and FFMPEG library. This drive was specifically created for this class with all the tools necessary to create a computer vision pipeline.

Library Liaison

Anamika Megwalu, anamika.megwalu@sjsu.edu

Course Requirements and Assignments

There are six programming assignments in this course. The programming assignments are designed as building blocks which lead to a culminating final project: web-based, computer vision face detection system. Detailed instructions and specifications for each programming assignment will be posted to my SJSU faculty webpages at http://www.cs.sjsu.edu/~bruce/ with adequate time for students to complete each assignment by the assignment deadline. Students are strongly encouraged to ask the instructor for clarification on each programming assignment specification. There will be an in-class oral presentation and demonstration of the final project as well as a written final project report. Lastly, there will be on midterm and one final exam.

Final Examination or Evaluation

There will be a final exam on **Monday, May 22 in SCI-311 at 7:45PM**. The final exam will be based on in-class lecture material and material from our textbook. The final exam will be comprised of 15 multiple choice questions. This is a closed note, closed-book exam. The final exam is worth 15% of your overall grade.

Grading Information

Each programming assignments will include detailed specifications. The programming assignments will be graded collectively as a group. This means everyone in the group gets the same grade for

each programming assignment. I expect each of your programs to compile (if applicable) or interpret (where applicable) without syntax errors or logic errors. Points will be deducted for programs that do not meet the specifications and/or programs containing syntax errors.

The programming assignments will be due upon completion of the project on May 15, 2017. Only one member of the group needs to submit the programming assignments to Canvas. All members of the group will receive the same grade for these assignments.

The final project oral presentation is an opportunity to discuss your project and demonstrate it to the class.

Each member of the group is expected to speak for approximately about their contributions on the project. After each member of the group has spoken, the entire group will present a demonstration of the project itself (i.e. run the project).

Each member of the project will individually write and submit a final project report. This is an individual assignment, not a group assignment. The final project report should include a title page followed by a two-page report. The report should include the following:

- Your contributions to the project.
- An overall assessment of the success and failures (if any) on the project. This is an opportunity to discuss time management issues, communication issues among team members, etc. It's an opportunity for you to reflect upon the project and how well the team performed.

Incomplete assignments

Points will be deducted for incomplete or partially working solutions. Point penalties vary with each assignment but are listed in the programming project assignment.

Late assignments

Any assignment submitted after the specified due date will be considered late and subject to minimally 50% loss in points (additional points will be deducted for incomplete projects as noted above). For example, if programming project 1 is submitted late, it will be worth, at most, 2.5 points.

Makeup Exams

Exams are closed note and closed book. Exams must be your own work. Makeup exams will only be given in extraordinary circumstances with instructor approval; instructor MUST be notified in advance.

Grade breakdown

ASSIGNMENT OR EXAM	POINT VALUE
Biographical writeup**	5
Programming Project 1: Database schema*	5
Programming Project 2: Web-based user login*	5

Programming Project 3: Extract still images from video*	10
Programming Project 4: Determine facial landmarks*	10
Programming Project 5: Track eye pupils*	10
Programming Project 6: Web-based display results*	10
Final Project Oral Presentation**	5
Final Project Report**	10
Midterm Exam**	15
Final Exam**	15
TOTAL POINTS	100

* Graded as a group (each member of the group gets same grade)

** Graded as an individual

Grading Scale:

Percent range	Grade
97% to 100% inclusive	A+
93% to 96% inclusive	A
90% to 92% inclusive	A-
87% to 89% inclusive	B+
83% to 86% inclusive	B
80% to 82% inclusive	B-
77% to 79% inclusive	C+
73% to 76% inclusive	C
70% to 72% inclusive	C-
67% to 69% inclusive	D+
63% to 66% inclusive	D
60% to 62% inclusive	D-
Below 60%	F

Classroom Protocol

Regular class attendance is highly recommended. Students are responsible for knowing all materials covered through in-class lectures and assigned readings. Please be mindful of fellow students and the instructor by not talking on mobile phones during instruction. Students are expected to leave the

class quietly in the event they must use their mobile phones.

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' Syllabus Information web page at http://www.sjsu.edu/gup/syllabusinfo/

CS-160-03 / Software Engineering, Spring 2017, Course Schedule

Course Schedule

Week	Date	Topics, Readings, Assignments, Deadlines
1	Monday, January 30	Introduction: Course goals and objectives. Lecture: What is software engineering? Readings: Chapter 1 of *Beginning Software Engineering*
1	Wednesday, February 1	In-class assignment: Your programming experience and biographical information
2	Monday, February 6	Lecture: Postgres RDBMS (Relational Database Management System)
2	Wednesday, February 8	Lecture: Document management Readings: Chapter 2 of *Beginning Software Engineering*
3	Monday, February 13	Lecture: FFMPEG library and systems programming tools
3	Wednesday, February 15	Lecture: Project management Readings: Chapter 3 of *Beginning Software Engineering*
4	Monday, February 20	Lecture: Processing FORM data from an HTTP POST
4	Wednesday, February 22	Lecture: Requirement gathering Readings: Chapter 4 of *Beginning Software Engineering*
5	Monday, February 27	*Conceptual design reviews and prototype demos*
5	Wednesday, March 1	*Conceptual design reviews and prototype demos*
6	Monday, March 6	Lecture: High and low level design Readings: Chapters 5 and 6 of *Beginning Software Engineering*
6	Wednesday, March 8	Lecture: Secure session management with HTTP
7	Monday, March 13	Lecture: Programming the Postgres C API (Application Programmer Interface)

7	Wednesday, March 15	Lecture: Development Readings: Chapter 7 of *Beginning Software Engineering*
8	Monday, March 20	**MIDTERM EXAM**
8	Wednesday, March 22	Lecture: Active Shape Modelling (ASM) and Active Appearance Modelling (AAM)
	Monday, March 27	SPRING BREAK: NO CLASS TODAY
	Wednesday, March 29	SPRING BREAK: NO CLASS TODAY
9	Monday, April 3	*Application design reviews*
9	Wednesday, April 5	*Application design reviews*
10	Monday, April 10	Lecture: Testing Readings: Chapter 8 of *Beginning Software Engineering*
10	Wednesday, April 12	Lecture: Eye pupil tracking
11	Monday, April 17	Lecture: Deployment, Metrics, and Maintenance Readings: Chapters 9, 10, and 11 of *Beginning Software Engineering*
11	Wednesday, April 19	Lecture: Drawing with OpenCV
12	Monday, April 24	Lecture: Delaunay triangles
12	Wednesday, April 26	Lecture: Predictive models Readings: Chapter 12 of *Beginning Software Engineering*
13	Monday, May 1	*Code reviews*
13	Wednesday, May 3	*Code reviews*
14	Monday, May 8	Lecture: Iterative models Readings Chapter 13 of *Beginning Software Engineering*
14	Wednesday, May 10	Lecture: RAD Readings Chapter 14 of *Beginning Software Engineering*
15	Monday, May 15	*Application demonstration* **DUE: Programming Assignment 1** **DUE: Programming Assignment 2** **DUE: Programming Assignment 3**

		DUE: Programming Assignment 4 **DUE: Programming Assignment 5** **DUE: Programming Assignment 6** **DUE: Final Project report**
FINAL EXAM	**Monday, May 22**	**FINAL EXAM at 7:45PM in SCI-311**