**CS3337 Syllabus - MW & Saturday**

**Instructor:** Dr. Jose M. Macias (Call me Jose)

*email: [josemasia@gmail.com](mailto:josemasia@gmail.com) [jose.m.macias@jpl.nasa.gov](mailto:jose.m.macias@jpl.nasa.gov)*

**Jose’s Hours:** Phone #s: work (818)393-0771
By appt only. Mo-Th: 5-6pm. Fri: 9am – 6pm. Sat 9-12pm.

**Alternate Instructor:** Richard Cross

*email: [richard.csdept@gmail.com](mailto:richard.csdept@gmail.com)*

**EMAIL HEADING:** Your email subject line SHALL include
“cs3337 - (Meeting days) MW or SAT - The Subject”

**Office hours:** Make an appointment by email/phone (Jose or Richard).

**Course number** CS 3337

**Course name** Software Design

**Credits** 4 units

**Contact hours** 5 hours/week

**Coordinator** Prof. Raj Pamula

**Pre-requisite:** *Very Good* programming skills using Java, or C++, or C or ?

**Text books**

ISBN-10: 1284106004

**References:**

“Software Engineering”, Ian Sommerville, Addison Wesley

**Course Information**

Catalog Description: Methodologies and tools for requirements analysis and design of large complex software system, process models, project planning, tracking, documentation, communication, and quality assurance; group laboratory project; oral and written presentations. Lecture 2 hours, laboratory 3 hours. Graded ABC/NC.

**Prerequisites:** CS 2013 “Programming with Data Structures”.

**Required/Elective:** This course is **required** in the BS program.

**Course Goals**

The Student Learning Outcomes (SLO) addressed by the course are:
SLO #5. Students will have the training to analyze software engineering problems and identify and define the computing requirements appropriate to their solutions.

SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.

SLO #7. Students will be able to communicate effectively orally and in writing.

Other outcomes of instruction:
At the end of the course, students are able to
- Estimate the cost and effort for software projects
- Make schedules for software projects
- Elicit software requirements
- Create data model, flow-oriented model and behavior model
- Convert the requirement models into software architectures
- Implement component-level design

Brief list of topics to be covered
- Analysis of large Software Engineering Projects using models to estimate the Software Project effort and budget.
- Software Process
- Requirement Engineering
- Elements of Design Engineering
- Project Implementation
- Project Presentation and Defense
- Project Version Control using Git and GitHub

Laboratory Projects
The students will be randomly divided into groups of 4 students per team. The students will complete a project based on a selected topic from their survey or provided by the instructors. One group member uploads the project under “Software Design and Implementation Document” assignment on CSNS. Projects will include steps such as:
- Build data model, flow-oriented model
- Behavior model for the selected project
- Design software architecture based on the requirements
- Create the component-level design
- Implement system components
- Integrate system components
- System Demo and Presentation

Format of the Software Requirements Document/Software Design Document (SRD/SDD)
The instructors will provide sample of the Department’s approved Requirements and Design Documents Templates for the documents that will be created during the semester.
Grading

To receive a grade in this class each individual part must receive at least a passing grade before the grades are weighted, that is, Exams, Quizzes, Readings, Documentation and Peers Evaluation all must pass independently BEFORE they can be weighted. See below the grading curve, any grade at or below 50% is considered a non-passing grade.

Exams

There will be two exams. The exams are open book/notes. The exams are weighted 15% (Midterm), and 30% (Final) of the total grade.

Quizzes on Readings
(classical papers on Software Engineering)

Readings are a most fundamental part of the class. Readings are controlled by means of quizzes. The minimum expected number of readings is 6. All readings are required. The average of all quizzes is 20% of the total grade. Readings may include:
- At least 4 additional selected readings.

Project Grading Policy

The “Software Requirements Document, SRD” and the “Software Design Document, SDD”, will be evaluated at 15% total.

The last important grade is the peers evaluation grade at 20% of the total grade. This grade is the average of the grade each member of the team assigns to the other team members. This grade is sent in a private email to the instructor. It reflects the opinion of the team members regarding the contribution, work and dedication the other team members have put in the execution of the team's project.

Mandatory attendance

A large portion of this class revolves around in-class project development. Therefore, this class requires 100% attendance. In addition, the students must be on time after each break and at the beginning of class. Three “late arrivals” will account for a full day of missed classes. Absences more than a full week of classes will receive a final grade "NC". Emergency and/or well identified unexpected circumstances will allow for at most missing one week of classes. Still, this situation may impact the student's grade at least one step down.

Academic Integrity
(University Policy)

The California State University has issued strong warnings disqualifying cheating by the students. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of “NC” for the course and will be reported to the Department Chair for further disciplinary action.
Grading Curve
(Highest class grade = 100%)

A : 90% or better
A- : 80% or better
B+ : 70% or better
B  : 60% or better
B- : 50% or better
NC : below 50%

ALL STUDENTS IN THIS CLASS WILL BE ASKED TO SIGN A DOCUMENT (RESIDING IN CNS, POSSIBLY … ) CALLED “SYLLABUS UNDERSTANDING” ASSERTING THAT THEY HAVE READ AND UNDERSTOOD THIS SYLLABUS AND ALL ITS PRE-CONDITIONS FOR THIS CLASS.