CSCI372 Syllabus

Study the generic rules for all my classes [../syllabus.html] as well as the rest of this [index.html] web site.

Course Information

COMPUTER SYSTEMS IN ORGANIZATIONS

General system theory, information theory. Modeling organizational activities, hardware and software with the Unified Modeling Language. Human-computer interaction, systems engineering. Students will study part of an actual or hypothetical organization. Prerequisites: CSCI 201, MATH 272, MATH 262. (4 units)

I will waive the MATH requirements this quarter.

This course has been taught at least 6 times. It is normally scheduled once per year. There is no required text book. Instead there will be readings on the Web. I will refer you to a particular page on my web site for this class [http://csci.csusb.edu/dick/cs372/] for you to study. You will have to follow some of the links on these pages and study what you find there. Most readings will have a set of review questions to be answered as the assigned work for the class.

Why Take This Class?

1. The course is required in the BA interdisciplinary and web options.
2. Your long term career depends on how well you understand the way organizations work and how computers can be used in them.
3. It is important to produce software that helps an enterprise work better.
4. The other parts of a system have to work for the software to be useful. The software must fit the other parts or there will be trouble. Example: My HMO/Insurance is highly computerized but
   1. My doctor complains that computerizing his paper work has reduced the number of patients he sees each day from 30 to 20... and he is working harder than before.
   2. In another part of the organization, the pharmacy, you can order a refill by the web or the phone, and it appears to work, but the order does not get through to the pharmacists. Sometimes it gets through, and is fulfilled and then lost!
   3. My wife has to give an annual chest xray report to the local school district. This usually works ok but one year the filled-in form vanished between the XRay tech, her doctor, and the reception desk where it is picked up. It took two hours to find it.
   4. The HMO has got an excellent private Email system for doctor-patient-laboratory communications. But some parts of the enterprise do not use it and rely on paper and phone calls.

   A computer professional should be able to find out why that above stories are true and what to do about them.
5. This course reviews hardware and software options from the point of view of both the user and the computer professional.
6. Knowledge of systems analysis and design will count in winning a job. This is course covers a lot of systems analysis and design.
7. Being able to spot risks, opportunities, and problems is a vital skill. This course will show you how to look at a situation and find them. It also has ways to discover improvements that reduce risks, take advantage of opportunities, solve problems, etc. This is a useful set of skills in any job.
8. We will cover the traditional and Unified forms of diagrams.
9. I will show you how to design and evaluate systems of programs.
10. I will suggest methods for planning and managing projects.
11. We will cover the operation, support, and maintenance of systems. These are a vital activities in real systems.
12. Field Trips!
13. No Programming!

Note: many of the skills are embodied as procedures, calculations, and diagrams: Data Flow Diagrams, Critical Path Method, Entity-Relationship Diagrams, Normalization, Activity Diagrams, etc., I expect you to learn to do them correctly.

Reading

(Required Reading): You will be required to read handouts and designated pages on the World Wide Web. Detailed reading assignments will be in the pages for each class on the web -- these will be ready in time to prepare for each class.

Here is a list of topics covered
1. Analysis: Theory of Systems and Fact Finding [a1.html], Organizational Context of Systems Work (People) [a2.html], Systems Architecture (Hardware+Software) [a3.html], Modeling the Data in a System (Data) [a4.html], System Modeling (Procedures) [a5.html]
2. Choices: Selecting Projects [c1.html], Strategic Thinking [c2.html], Project Planning and Management [c3.html]
3. Data: Human Interfaces [d1.html], Data Element Design [d2.html], Entity Relationship Design [d3.html], Detailed Data Design [d4.html]
4. Requirements: Processes, Procedures, and Logic [r1.html], Functions and Use cases [r2.html], System Design [r3.html]

As you study these web resources look out for Stories, Patterns, Principles, Diagrams, Calculations, Errors, Smells, and Jokes.

Patterns are solutions to complex problems that are good enough to be used again and again. Watch out for these in each topic we cover.

Principles are general properties or rules that we use. Some are based on theory and sometimes on practice. Again look out for these.

Diagrams are useful tools. Typically we sketch them on paper or a board as we think about the systems. We use tools to make tidy versions when we need to either preserve or present them. We express our ideas about particular systems using pictures with well defined meanings. An example is the Data Flow Diagram or DFD that shows how parts of a system effect each other. Another is the Activity Diagram that records a particular procedure used in an organization. You must master the rules for half-a-dozen diagrams so you can use them as needed in your career. We use these like a mathematician uses formulas and an engineer uses circuit diagrams (eXtreme Programming uses them for requirements. "War stories" of real life systems are worth taking note of. I will also use every day examples (making coffee, say) to illustrate techniques that are normally applied in more realistic situations. Also see Jokes below.

Calculations help check that our diagrams work. We use calculations to evaluate designs and plans. For example we can use an activity diagram to describe the work to be done next week, and the "Critical Path Method" to calculate how long it will take and which steps in the plan are most critical for completing it before the week is over.

Errors occur when you don't follow the rules for particular Diagrams or Calculations -- like when you add 2 and 2 and get the answer 5.
1. 2 + 2 =
2. These are like syntax errors uncovered in your program by a compiler. You must avoid these (and spot them) to do well in this class.

Smells tell you something is wrong. For example a good doctor can diagnose a strep throat from the smell on the patient's breath. Similarly a program that does not give a name to a constant does not smell good: it might compile and run, but something is rotten in it. A smell in a diagram does not mean it is incorrect -- but it means you should try figure out what is really going on, and perhaps change the diagram.

Jokes encapsulate folk wisdom. Here is an example. [TireSwingTable.html] (Cartoon) [treeswing.htm] (History and origins)

Resources

This web site [http://www.jiludwig.com/] seems to talk sense.

The Wikipedia is fairly trustworthy on analysis and design, but check with me.

Instructional Methods, Work and Grading

I will be using my standard straight-line grading scheme based on a 500 point nominal total. See [../../syllabus.html] on my main web
Lecture Discussion

40 points(8%). Lectures and discussions will be based on assigned reading. Participating in the 20 classes will be worth 2 point each. My plan is to set you lots of exercises starting with a task to be done while I grade the [Assigned Work] next below. There will be short lectures.

Assigned work

36 points(7.2%). There are 18 assignments. Each is worth 2 points. You will normally be given some web pages to study. They end with a set of review questions. You should try to answer as many of review questions as you have time for. Bring one question and its answer to class on a piece of paper with your name on it. It is worth 2 points. If you don't have a review question+answer hand in a blank sheet of paper with your name on it to at least prove you were in class. In some class meetings there is no reading and so no review questions. Instead hand in a question you want answered.

Field Trips

45 points(9%). There will be 3 field trips (TBA). Two of these are "real" and one is "virtual" this quarter (Fall 2011). Participation is worth 5 points and a written report is worth 10 points. The report is due at the start of the following class. It should be at least one page long. It should present your own observations and thoughts.

You can make up for missing a field trip by organizing your own visit to any office on campus and writing a report on how computer technology is used in that office.

Project

50 points(10%). Parts of five(5) sessions (Schedule) will be dedicated to developing a team project that interests you. 5 points will be given for work done in class time and a further 5 points for completing the work ready for the following session. Hand in what you have at the start of the next session. I may also ask for a short progress report from each team at the start of the class. At the end of the course you will be asked to present your project to the class. These classes will be passed to CSCI375 for further development. In the past some projects have been implemented as Senior Projects.

Quizzes

150 points(30%), 15 points each in 10 classes(Schedule). Note. This the largest component in the assessment prior to the final. They are the best predictor of your score in the final and so the course. But you can improve your skills and knowledge before the final by reviewing the graded quiz. Quizzes will test your memory of topics covered plus skill with the techniques covered.

Bonus

You can earn the standard 5 points bonus/make up points by attending a computer science seminar or Masters project presentation. See [../seminar/] for the schedule. You must attend the seminar and, at the next class meeting, email me a short report: who, what, when, why, how, ... (less than 250 words) on the presentation within 24 hours of it finishing. You can make up 5 points lost before the final. They have no effect on your score in the final.

Course Work Max 300 points

You can score a maximum of 300 points in bonuses and course work.

Final Examination 200 points

200 points comprehensive(40%). Questions will be based on work done in class (including quizzes, field trips, and project work) and in assignments during the quarter. This final examination needs special preparation -- go back over everything we have done to prepare for it. See [mock.html] a template for a final and questions [finals.html] from previous year's finals.

Policies and Schedule

See my generic syllabus [../syllabus.html] for the general policies that apply in my classes. Also check out the on-line [schedule.html] on my web site.

Changes Can Happen

This is a tentative syllabus. If I have to change anything it will be announced on this course's blog [index.html], in class, or even by EMail to your CSUSB Email address.

What is this class about

This class includes materials often covered in courses called


. . . . . . . . . (end of section CS372 Syllabus) <<Contents | End>>

Abbreviations

1. TBA::="To Be Announced".
2. TBD::="To Be Done".

Links

Notes -- Analysis [ a1.html ] [ a2.html ] [ a3.html ] [ a4.html ] [ a5.html ] -- Choices [ c1.html ] [ c2.html ] [ c3.html ] -- Data [ d1.html ] [ d2.html ] [ d3.html ] [ d4.html ] -- Rules [ r1.html ] [ r2.html ] [ r3.html ]

Projects [ project1.html ] [ project2.html ] [ project3.html ] [ project4.html ] [ project5.html ] [ projects.html ]

Field Trips [ F1.html ] [ F2.html ] [ F3.html ]

[ about.html ] [ index.html ] [ schedule.html ] [ syllabus.html ] [ readings.html ] [ glossary.html ] [ contact.html ] [ grading/ ]

End