

CSC-4330

SOFTWARE SYSTEMS

DEVELOPMENT

Dr. Sukhamay Kundu

Computer Science Dept, Louisiana state University
Baton Rouge, LA 70803

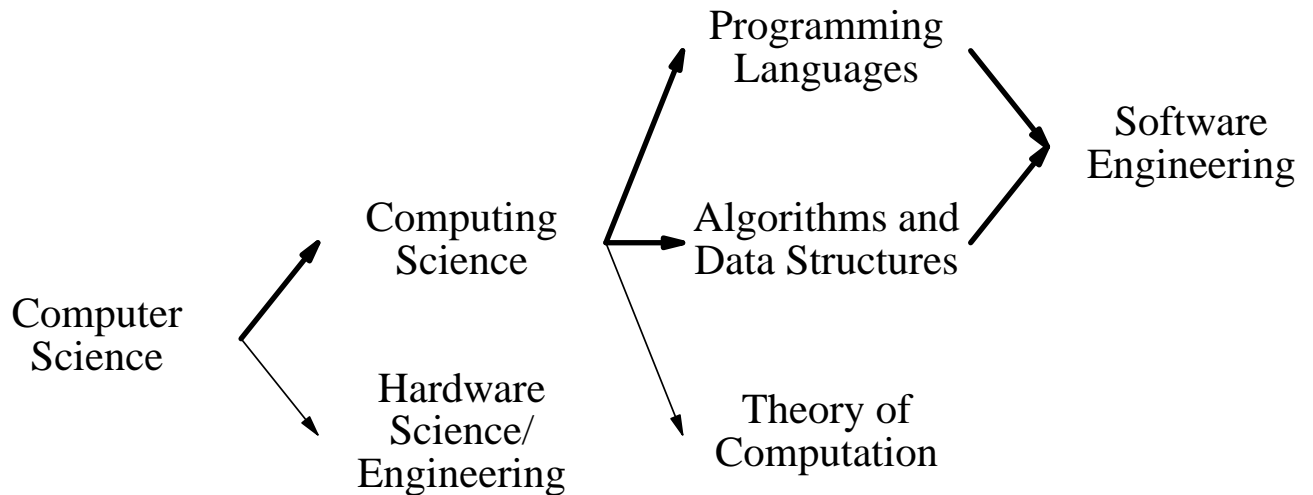
287 Coates, 8-2246, Off. Hours: 15:30-5:00 MW
kundu@csc.lsu.edu

Grader: John Burris, 8-1406, Off. Hours: 10:00-12:00, 162 Coates
jburri4@lsu.edu

Fall, 2009

(copyright@2008)

SOFTWARE: A FINAL END-PRODUCT OF COMPUTER SCIENCE



- **Theory of Computation:** What things are computable.
- **Algorithm Design:** How to compute those things efficiently.
- **Software Engineering:** Algorithm implementation to
 - Provide clear and simple user interface
 - Organize data and operations that simplifies understanding (including clear logic and clear coding style)
 - Use memory efficiently
 - Keep the computation efficient

The softwares and the hardwares are the two main final end-products of Computer Science.

HOW DO YOU KNOW A GOOD SOFTWARE FROM A BAD ONE

Question:

How will you decide whether a software (say, a text-editor or an email-software) that someone hands you out is a good one or not, i.e., whether to buy it or not?

- Ask "what is unique about it", i.e., its special features.
 - The features may depend on the type of software (text-editor vs. email).
- Experiment with some of the features or look at a demo.
- Credibility of the company that developed the software.
 - If the company is unknown, then you ask about the process employed in developing the software.
 - If you are a programmer, then you may even take a peek at the source-code.
 - Find out what sort of quality control/testing was used in developing the software.
- Find the price of the software and look at the reviews (if any).

Goals vs. Features:

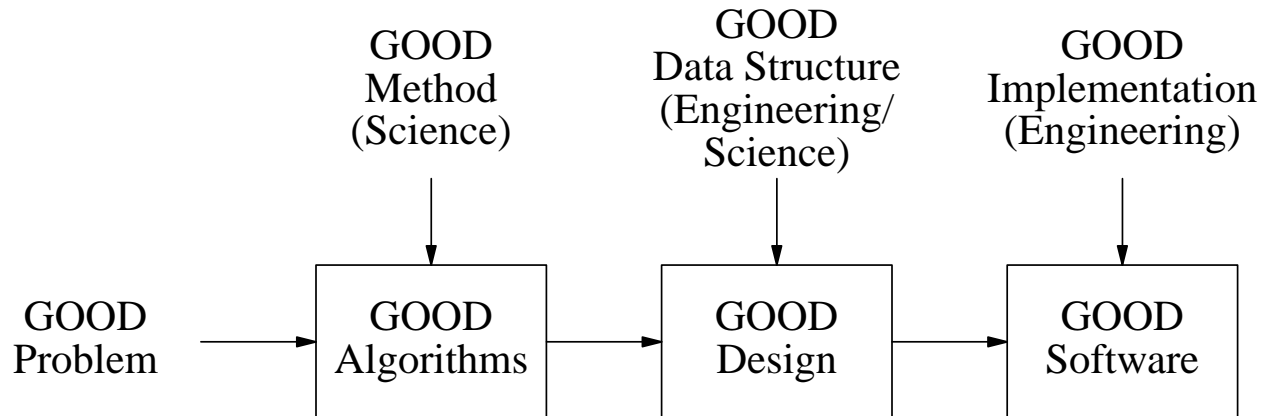
Goals: text-editor vs. email software.

unique features vs. cheaper/faster/more reliable

Features: They come from the requirements.

WHAT GOES INTO A GOOD SOFTWARE

- A lot of good things go together to make a good program.



Question: What makes a problem (software product-idea) good?

- The software-product would provide *useful/quality* service(s).
- Would provide an *integrated* set of closely related services.

Services to be provided are
the high-level Software Requirements.

Question: What can make a good software fail in the market place?

Question: What goes into *turning* a good software product-idea and good algorithms, etc into a market-successful software?

A DESIGN THAT WORKS IN ONE CASE BUT NOT FOR ANOTHER

- A design which is good for stores in a small shopping-small but is not so for rooms in a house:



- Separate entrances allow independent store-owners and to have independent store-hours and open/close operations.
- Linear arrangement makes it easy for customers to visit multiple shops in the same shopping trip.
- What makes this a bad design for rooms in a house?

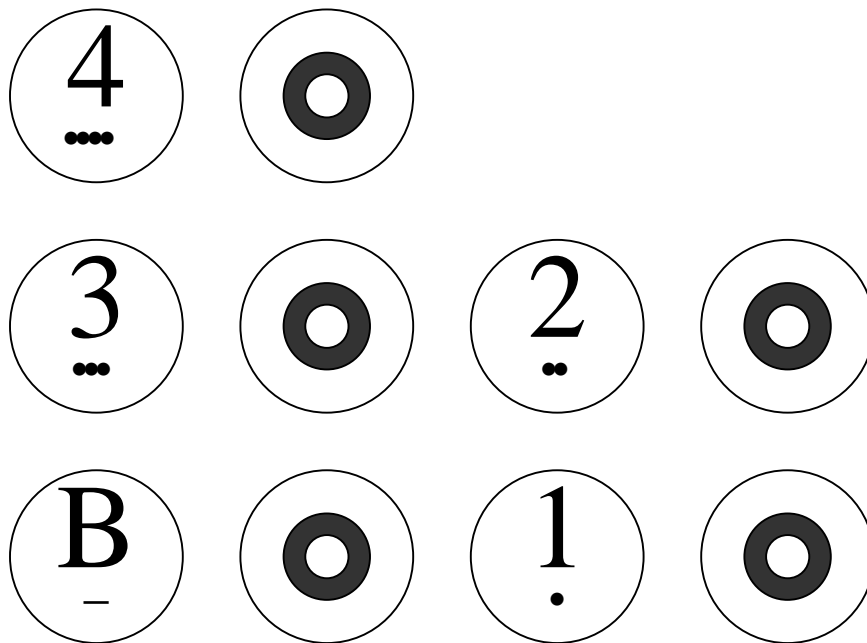
Different goals/requirements may call for different designs to meet those goals/requirements.

Question:

- ? Look at the placement of the elevator control-panel inside each of the three elevators at Middleton Libray. Do you see any problem? State a requirement for the control-panel placement that could prevent the problem if the placement was done accordingly.
- ? Give a panel design that could serve all three elevators (instead of a separate panel in each elevator)? Why this is not done?

A BLIND-PERSON ACCESSIBLE ELEVATOR CONTROL-PANEL DESIGN

- Each button with a thick smaller circle has a red-light in the center which turns on when you press the numerical-button left of it.
- Each numerical button has the proper brail-marks to indicate the number.



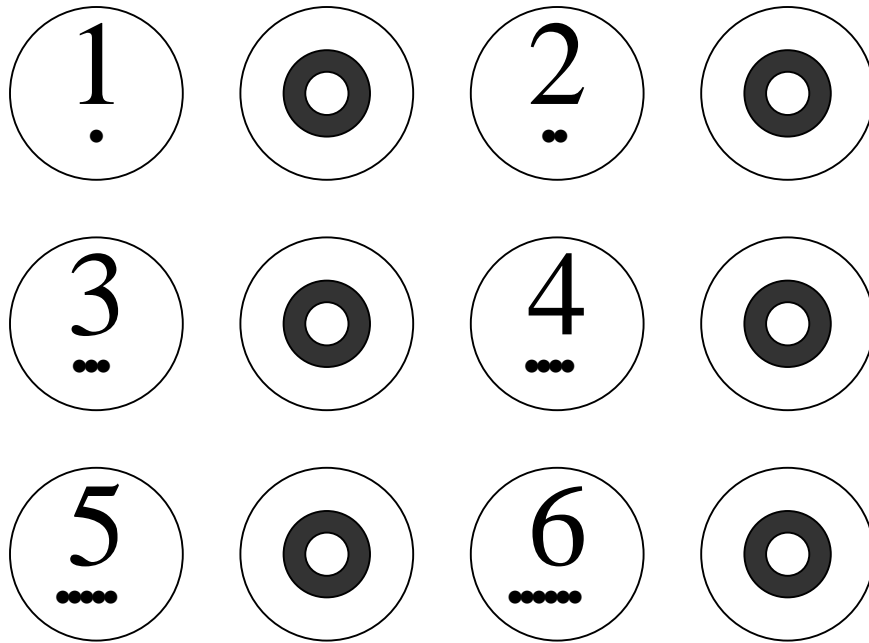
... other buttons ...

... other buttons ...

Question:

- ? What problems do you see in the placement of light-buttons – how can this cause problems for a blind person?
- ? Why only two numbers per row? Is it proper for the numbers to decrease left to right and top to bottom?

ANOTHER CONTROL-PANEL DESIGN FOR A MORE RECENT ELEVATOR



... other buttons ...

... other buttons ...

- Here, the numbers increase downwards!

WHAT IS INVOLVED IN DESIGNING

Problem Parameters vs. Design Attributes: Need not be the same.

Values of the parameters (more generally, constraints involving them) define the design problem.

Values of the attributes specify the design solution, and let us compare different solutions.

A Simple Design Problem:

Design a rectangle that *maximizes* the area with given perimeter p and width $\geq w$.



- Designing a rectangle = choosing design-attributes' values. Which of the following are suitable design-attributes sets (why)?
 - Area (or perimeter)
 - Length and area
 - Height and area
 - Length and height

Design Objectives or Requirements:

- Requirements are based on how (not why) the designed entity will be used; they provide a criteria for design evaluation.
 - A vegetable garden (types of vegetables and quantities, not what dishes will be cooked and for how many people)
 - A building plot (building type and size, space around it)
 - A park (different game-facilities and number of uses)
- Requirements give rise to the design constraints.

Software design is difficult in both aspects: defining the problem parameters and the design attributes.

EXERCISE

1. What kind of constraints can make the rectangle-design problem difficult?
2. Will a circle design problem be simpler than a rectangle design problem (if so, why)?

Stating a problem (requirements) in a way that helps the choice of design attributes and their values can be difficult.

SOFTWARE ENGINEERING

Software Engineering:

- A systematic method for structuring and building a software, specially large and complex software.

Role of Structure:

- Any large and complex artifact (a book, a bridge, a city, a software, a public garden, a company, a lecture, a picture, etc.) is built from smaller pieces, and the same is true for softwares.
 - The first building-step then is to define a suitable "structure" of the artifact in terms of its parts and their relationships.

The nature of the parts and their relationships can greatly differ from one type of artifact to the other.
 - The structure can be dynamic, with dynamic relationships that change with time. It may also have dynamic parts.[†]
- The structuring reduces complexity and hence errors, with each part being simpler than the whole and thus providing a better understanding of the artifact.

Question:

- ? Give an example of artifacts in a program-code with static parts and dynamic relationship among them.
- ? Give an example artifact with dynamic parts and static relationship among them.

[†] A dynamic entity or relationship has one or more dynamic attributes, which may be a part of the key. A static structure may have dynamic parts but no dynamic relationships.

IS THERE A SCIENCE BEHIND SOFTWARE ENGINEERING

Science:

- The particular types of structures that are useful in modeling Software (Information Processing and Computational).
- The methods for analysis of such structures.

Science of Data and Operations:

- Modeling data and their relationships.
- Modeling operations and their relationships to data.
- Analyses of these models and optimization to reduce complexity.

Engineering:

- Making use of such structures and analysis to improve the quality of a software.

Designing:

- A part of engineering.
- For a software, which is made of two main parts – data and operations, the design has two primary parts:
 - design (organization/structure) of operations, and
 - design of data.

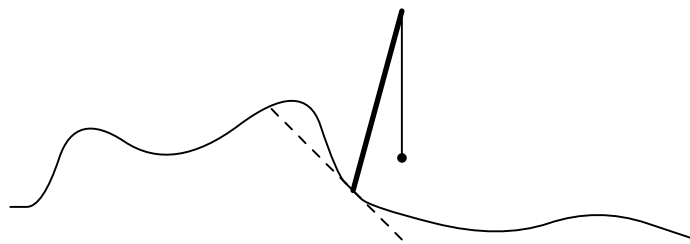
SCIENCE vs. ENGINEERING

Science of Gravity:

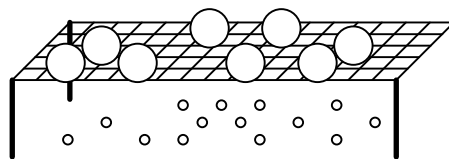
- The earth's gravity pulls every thing towards the center of earth.
Thus, the center of gravity of a weight attached to a string and hanging freely from a point lies vertically below that point.

Two Engineering Applications of Gravity:

- Putting a vertical pole. We can test whether a straight pole is vertical or not by hanging a small weight from the top of the pole. The pole is vertical if the weight just touches it.



- Separation of different size particles/grains using a strainer.



Question:

- ? If the pole is not straight, then what is the meaning of it being vertical (and how to test it)?
- ? If there are three different size particles, we need two different size strainers to separate them. If the total weight of each size particles are the same, what is the best way to arrange the strainers (i.e., the best separation process or algorithm)?

WHY IS ENGINEERING DIFFICULT

Choosing Proper Science to Fit The Problem:

- How to separate equal size particles of different materials?
 - Use other physical or chemical properties.

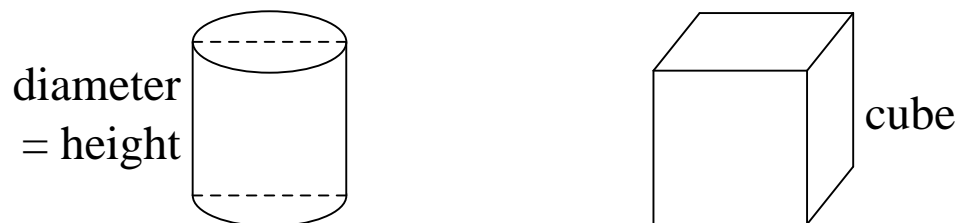
For example, we can separate iron and wood particles using the magnetic property of iron. Is there another way?

Use of Optimization to Choose Proper Shape, etc:

- The optimal shape of a soup-can for a given volume V :
 - For a cylindrical-shape can, height = diameter.
 - For a box, all sides are equal; its surface-area is, however, $4/\pi$ times larger than that of the cylindrical-shape can.

Cylindrical-shape metal soup-cans save the cost of metal. Why does the diameter almost never equal the height?

 - The large shipping packages are mostly of box-shape (not cylindrical), but they are rarely a perfect cube - why?



Balancing Utility and Cost:

- Often a major task in engineering is to choose a solution that meets (approximately) several competing requirements.

Designing A Solution (a part of engineering):

- Choosing among many possible combinations of alternatives.

EXERCISE

1. Although the optimal shape of the printing surface-area is a square, most printed-pages are of rectangular shape (not a square). Why?
2. Why do the boxes for copy-machine papers have different size than the boxes for packing clothing (wardrobes)?
3. Why does large cooking oil bottles/cans have handles and large soup-cans (or juice-cans like V8) does not?

WHAT IS NEW IN THIS COURSE

- You will define the problem for your project.
 - The emphasis is not clever algorithms as in a Data-structure or Algorithm design course.
 - Nevertheless, use them whenever possible.
- You will work in groups from start to finish, as in a software professional environment, with considerable effort spent in:
 - Reaching consensus within your group in all stages
 - defining the problem.
 - designing the overall solution approach.
 - creating the solution (clean and elegant code).
 - testing and demonstrating the solution.
 - Producing documentations
 - Clearly stating your ideas and the key concepts
 - Justifying (benefits of) your decisions/choices
- You will learn key issues in large scale software development and relevant modeling and analysis/solution methods.
 - Requirements, its analysis, and use in high-level design
 - Finite-state modeling, Workflow-modeling
 - UML and other related modeling
 - Software test-planing and other management issues

Question: How well does this fit your expectation of the course?

GRADING, EXAMS, ETC.

Grading: *A* (85 - 100), *B* (75 - 84), *C* (65 - 74), *F* (0 - 65)

Exams, Homeworks, Presentations:

Three Quizzes: 30%	Weekly/bi-weekly proj. reports: 35%
Homeworks: 5%	Interim presentations (2): 10%
Class participation: 10%	Final presentation: 10%

Programming: C/C++, JAVA, .NET, HTML, etc

Forming Groups (3 to 5 persons per group):

- Settle groups and the projects by the second week.
- Subsequent changes are possible only in extreme cases.

Other Industry Speakers:

- Several (approx. 3 to 5) through the semester.

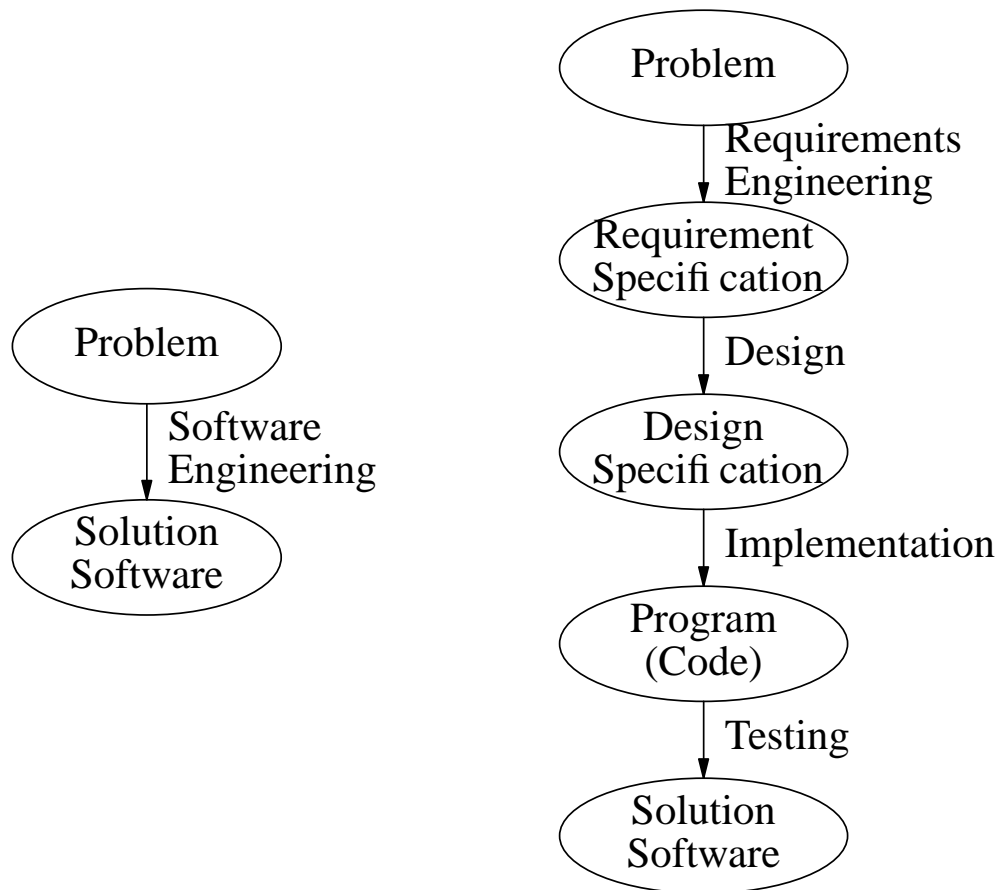
Submission for Next Class-period:

- Each of you will first prepare your own 1-page project description (title, project, problem, etc). Then, work together to prepare a group-version, and submit all the individual-versions and the group-version together as a package.
- Also, submit a document (group-version only) listing the major functions to be provided by your software/web and the major data items needed as inputs for these functions. Give a short description of each function and data item. Make a table to show for each function its inputs and outputs (there may be only limited inputs/outputs for web-related functions).

SOFTWARE ENGINEERING

Software Engineering (is more than programming):

- Methods for systematic development of large software, starting from the problem analysis to the delivery of well-designed and thoroughly tested software, including all documentations.



Waterfall process model.

Verification: Make sure that your goal/target is correct, i.e., you are going for the correct system.

Validation: Make sure that you achieved the goal/target system.

SOFTWARE ENGINEERING IS DIFFERENT FROM TRADITIONAL ENGINEERING

Differences:

- Natural laws of Physics/Chemistry does not apply to software in terms of structuring/organizing information and operations.
- Simple linear scaling does not apply from small software to large software.
 - Software complexity is not a linear function of the number of variables or the size of the code.
 - Software cost is not a linear function of its size.

Typical Questions One Can Ask of A Software:

S O F T W A R E
(Library system, text editors and text processing, email system, MATLAB, etc)

- What does it do, what problems does it solve?
 - Automation of library system allows to determine how many books a person has checked out at any time.
- Who developed it, how much did it cost, how long did it take?
- Is it stand alone? What OS does it run on?

Question: Which phases in the waterfall-model relate to which of these questions?

PROJECT-STATEMENT

Five Components:

- (1) *The title.* (It should be short and should help us focus thoughts and expectations.)
- (2) *The project.* (Describe what this project intends to do. It may give a hint of the larger problem of which the project might address a small part.)
- (3) *The problem.* (Explain the broader issues your project will help to solve, maybe only a part.)
- (4) *The impact.* (Tell specifically what your project will achieve, who will benefit and where will the impact be felt – the objective component.)
- (5) *The motivation/inspiration.* (Tell why this project is important to you personally – emotional component.)

Examples of Project Title (max 50 characters):[†]

<p>T_1: One Million Kids Safe: One PC at a time.</p> <p>T_2: For the Children - Feed the Body, Educate the Mind.</p> <p>T_3: Science for Indigenous Children.</p> <p>T_4: Wild Weather Fun.</p> <p>T_5: Creating Futures.</p> <p>T_6: EasyTutor - Free, Unlimited Online Tutoring Access</p>
--

Question: Choose a title and write a short (3 to 5 lines) for "The project" part.

[†] From American Express' members-project competition, 2008.
Length restriction for other parts: 500 chars each.

WHAT IS IN A PROJECT TITLE

Project Title:

- It should be short.
 - It is a very short (one phrase) summary.
 - It must avoid all unnecessary words.
- It should clearly indicate what to expect in the rest of the writing.
- To express something in a few words require identifying the most important issues and that requires a clarity of understanding.

Question:

- ? What is wrong with the following title of a published article? Which words can be taken out without losing value? Does it use capital letters in a systematic (logical - what is it) way?

Web Modeling Language (WebML): a modeling language for
designing Web sites

- ? If we keep all the words, can you reorder them in a way to better convey what the contents of the article might be?

EXAMPLES OF THE PROJECT-PART

- T*₁. The 1 Million Kids Safe National Campaign at its heart is a grass roots initiative to empower caretakers on an intimate level and educate them on the importance of protecting our nation's greatest investment, our children. The 1 Million Safe Campaign Website will act as the conduit between community, state agency and federal government with regards to child internet safety. Lastly, 1 Million Safe will host awareness rallies for youth around the country, promoting online responsibility.
- T*₂. My idea is to provide the needy children of the world with food for the body and knowledge for the mind. Under this project, children in under-served areas of the world will have healthier lives through nutritional supplements. In addition, internet-enabled laptops made just for children will provide access to educational mediums such as online classrooms, peer to peer interaction, and the limitless knowledge of the World Wide Web.
- T*₃. It will incorporate science education into the idle time and play arena of preschool and early primary school children in the small remote, relatively stable and peaceful indigenous communities in Northern Philippines. It will create a non-competitive context and apply a compassionate teaching approach -- congenial paths to joyful learning; it will capitalize on the bounty of surrounding nature which offers a ready-made holistic setting and subject material for natural science.

CONTD.

- T*₄. I believe there should be a center devoted entirely to the education of weather. Weather can be presented in such a way to be educational yet a lot of fun. Informal hands on education will reach more children than someone standing in front of them preaching about the weather. There is a lot of folklore surrounding weather that would be fun for the visitors to learn about as well as the science and technology used in predicting weather.
- T*₅. Creating Futures will provide pathways to employment through personalized computer training and certification. This project will establish a non-profit organization with the primary mission of helping transition U.S. veterans, individuals with disabilities, youth-at-risk and dislocated workers build valued, productive and rewarding jobs and careers in technical fields.
- T*₆. A website that connects students from around the nation to quality volunteer tutors through a live chat program. Students can chat individually with tutors as well as draw on virtual whiteboards and even use microphones to orally communicate back and forth. All tutoring will be free and available as long as tutors are online to take tutoring requests from students. The website should also offer an online community for students to interact on many academic topics and share knowledge and support.

Question: Rank (scale 1 to 10 = best) the project-statement in relation to their titles; justify your answer.

COMPLETE EXAMPLES OF PROJECT

The title: Innovation Breeds Innovation

The project:

A Web Site called the "Idea Exploration Portal" will help innovators (creators, composers, designers, inventors, researchers, leaders, etc.) in all fields share ideas and explore new ideas. The Idea Explorer (user) will dialog with a friendly avatar (Artificial Intelligence based) called 'SOFIA' who will serve as a tour guide in the jungle of innovative ideas.

The problem:

Innovation is the most important solution to the challenges faced by modern powers such as the United States, Europe, Japan etc. who are outsourcing "less critical" parts of their economies to developing countries in Asia and South America for example. Also, the global scale and complexity of the issues we face today, require new ways of thinking, in more creative ways than ever before in our history

The impact:

The "Idea Exploration Portal" will unify and focus the innovation community spread over many disciplines and areas of activities, that are currently fragmented, isolated. This isolation prevents cross-disciplinary fertilization of ideas.

The motivation:

20 years + of research experience at NASA and other research centers, helped me see the need for new tools addressing the creative process directly. No such tools exist, even though the need for innovative ideas is critical to solve the complex issues we face today.

Question: Do you need to mention your skills to do the project?

CONTD.

The title: Fulfilling America's Promise

The project:

Develop a tutoring program for high school students at risk of dropping out, focusing on hard-hit urban areas. However, instead of adults tutoring the high school students, the high school students would tutor younger students. Research has shown that being involved in meaningful service-learning can improve academic performance and student behavior. Moreover, by tutoring, the high schoolers not only have a responsibility to themselves, but also to their younger counterparts, to study.

The problem:

In America, nearly one-third of all public high school students will not graduate with their class. For African Americans, Hispanics, and Native Americans, the number is one-half. In Detroit, one of the hardest hit areas, the dropout rate is 75%. High school dropouts are eight times more likely to be in jail, twice as likely to slip into poverty in a single year, and three times as likely to be unemployed than high school graduates. This is not a "problem", this is a crisis.

The impact:

I expect this project not only to reduce the dropout rate in some of the nation's hardest hit areas, but also to improve the outlook of thousands of American students who might otherwise dropout. By keeping these students in school, this project would increase their average income and employment rates, ultimately increasing tax revenues, improving the economy, and reducing the strain on welfare programs. I also expect the program and its effects to expand.

The motivation:

Having just graduated from high school, I long knew students are dropping out at alarming rates. Last year, I had the opportunity to listen to a speech that discussed the mitigating effects that engaging at-risk students in service-learning has on dropout rates. Doing more research, I learned that it is one of the easiest and most effective ways to deal with this crisis.

CONTD.

The title: Easy Tutor - Free, Online, Unlimited Access.

The project:

A website that connects students from around the nation to quality volunteer tutors through a live chat program. Students can chat individually with tutors as well as draw on virtual whiteboards and even use microphones to orally communicate back and forth. All tutoring will be free and available as long as tutors are online to take tutoring requests from students. The website should also offer an online community for students to interact on many academic topics and share knowledge and support.

The Problem:

Many students struggle through studying and homework and often enter the classroom not understanding the content previously taught. Hiring private tutors is often very expensive and simply improbable for many students. The internet is easily accessible from home, school, a local library, or community center so offering free tutoring through a website can make it available to anyone needing help. Tutors will provide support in any subject, allowing the student user to gain a proper understanding.

The impact:

The project will have a widespread impact on schools and individual students around the world. Many students will now know that they have an alternative resource to look to when they feel that all hope is lost in school. Furthermore, the ability to accept volunteers as tutors will allow many who share a passion for teaching to do so in the comfort of their own homes through flexible scheduling. Many students themselves need service hours and volunteering here would allow them to teach too.

The motivation:

Having recently graduated from high school, I am aware of the need for supportive educational opportunities. I saw many of my peers struggling with no way out. They would come back unprepared to test, write an essay, or present a project. Had this project been in place, they would be able to go home and request the unlimited support of a tutor free of charge to make sure that they walk in the next day prepared and ready for anything. With this project, every single student would have a resource.

REQUIREMENTS/DESIGN FOR DISPLAY OF DATES & DAYS IN A CALENDER-MONTH

August						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

August						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

August				
S	5	12	19	26
M	6	13	20	27
T	7	14	21	28
W	1	8	15	22
T	2	9	16	23
F	3	10	17	24
S	4	11	18	25

January	February

March	April

⋮ ⋮

⋮ ⋮

January	July

February	August

⋮ ⋮

⋮ ⋮

- Why don't we have row-labels for the top two tables?

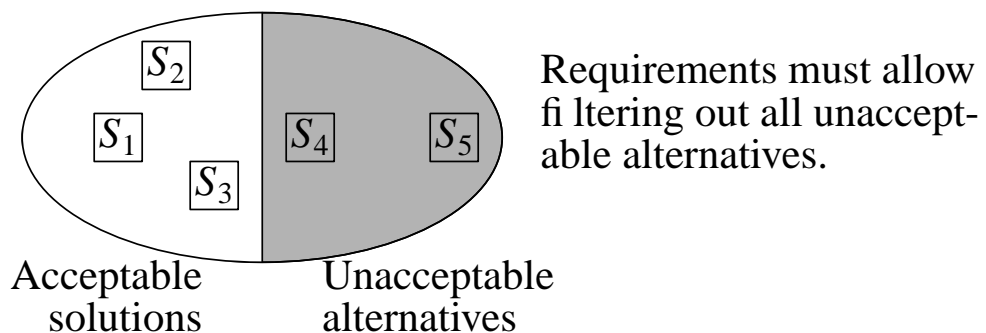
REQUIREMENTS: THE BASIS OF GOAL-ORIENTED SOFTWARE DESIGN



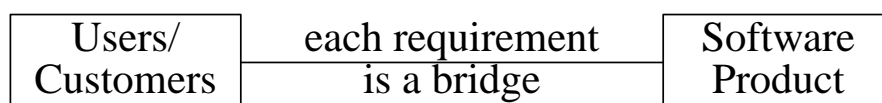
Question: What should be the output for erroneous inputs?

Requirements:

- (1) A set of criteria for the "external behavior" of the system ("what" vs. "how", internal behavior) of the system, that distinguishes the acceptable solutions from the others.
- (2) The requirements form the basis of acceptance-test; they must be testable/verifiable.



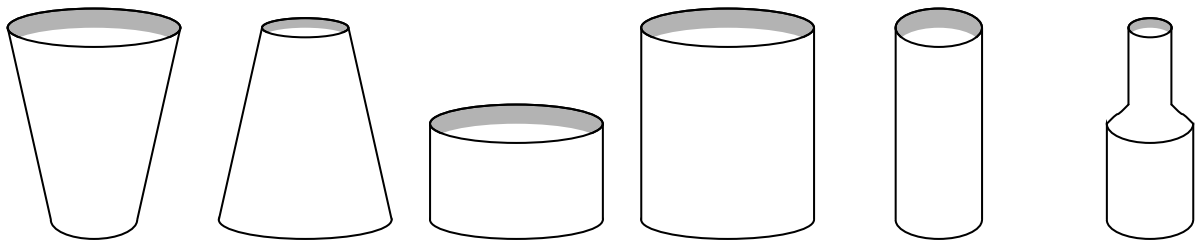
Question: Why do we emphasize here the "external behavior"?



REQUIREMENTS ARE DIFFICULT TO VISUALIZE AND TO STATE

Example.

- Some common and some uncommon designs/shapes of a simple cup for drinking-water (without the handle).



Question:

- ? State some requirements to filter out the uncommon shapes.
- ? Are there other requirements that are not perhaps "seen" from these examples?

Formulating requirements:

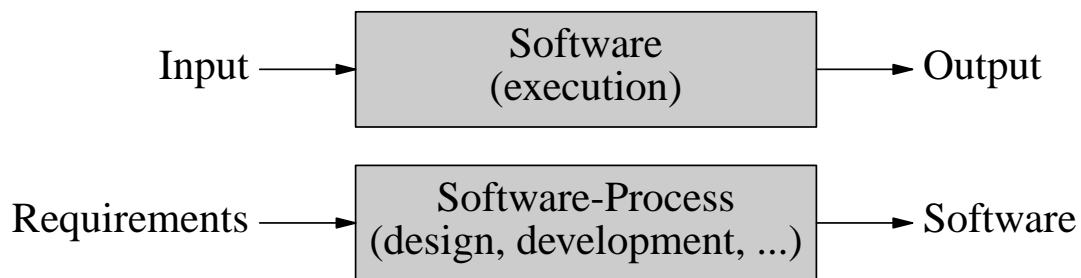
- Always use example cases as a guide.

Requirements: You get what you ask for.
Missed requirement = less useful final product.

REQUIREMENTS, SOFTWARE, AND SOFTWARE-PROCESS

Requirements: A high-level, black-box, user-view of software.

- Black-box: requirements come *before* software is built.
- High-level: requirements focus on "*what*" and not "how".



Functional requirements:

- What inputs, including applicable constraints, does it need?
- What outputs, including their properties, does it produce?
- What functionality, i.e., relationship holds between the inputs and the outputs?

Question:

- ? What are some (comparatively) minor issues of a software?
- ? What situations make us look at the requirements once again after building a software?
- ? What is wrong with the requirement "Write a software to classify triangles"? Give a correct form.

EXAMPLE OF REQUIREMENTS

- Suppose we have (or plan to develop) a software to count #(words in a text-fi le) and #(characters in those words).



Requirements:

- *Functionality*: Count #(words in a text-fi le) and #(characters in those words).
- *An input oriented requirement* (part of pre-condition):
There is no restriction on the word-size. (Alternatively, one could say that each word is, say, ≤ 20 characters.)
- *An output-oriented requirement* (part of post-condition):
 $\text{wordCount} \leq \text{charCount}$, both being 0 for an empty fi le.

Always Explain Requirements (with examples):.

- For the fi le shown below as a string and \boxed{e} for end-of-fi le, we have word/char counts = 5/22 or 6/21 (including ".").

This is a short text-fi le. \boxed{e}

Question:

- ? Give a requirement related to word-separators. Which category of functional requirement does this fall into?

TWO WordCharCount FUNCTIONS

- What would be some requirements that would make one or both of these invalid?

```
#define WORDLEN 20
void WordCharCounts(FILE *inFile)
{ int i;
  char word[WORDLEN+1];
  wordCount = charCount = 0;
  while (fscanf(inFile, "%s", word) > 0) {
    wordCount++;
    for (i=0; i<=WORDLEN; i++)
      if ('\0' == word[i]) break;
      else charCount++;
  }
}

void WordCharCounts(FILE *inFile)
{char ch;
  wordCount = charCount = 0;
  while (fscanf(inFile, "%c", ch) > 0)
    if ((ch != ' ') && (ch != '\n')) {
      charCount++; wordCount++;
      while (fscanf(inFile, "%c", ch))
        if ((ch != ' ') && (ch != '\n'))
          charCount++;
        else break;
    }
}
```

Question:

- ? How do you defend that none of these functions find the number of lines in the textfile? How could you fail to defend?

STATING REQUIREMENTS IS DIFFICULT

Requirements for WordCharCount-function:

Determine #(words in a textfile) and #(characters in those words).

How good is the above requirement-statement:

- Does it specify the input/output and their relationship?
 - What domain concepts are used to express the requirements?
- Is there any ambiguity in the requirement?
 - Is a punctuation mark (',') next to a word part of it?
 - Is "John's" one word? How about "don't"? Is "open/close" one word or two words?
 - How many words and characters are there in
 - "I don't know John's brother, but I know him."
 - What happens when a word is split between two lines as in
 - Mr. Johnson is an extra-
ordinary person.
 - What characters other than blanks and new-lines are considered word-separators?
- Is this requirements testable?

Determining requirements is not an easy task, but good requirements are a must for a successful software.

DIFFERENCE OF VIEWPOINTS IN REQUIREMENTS AND IMPLEMENTATION

Key Difference:

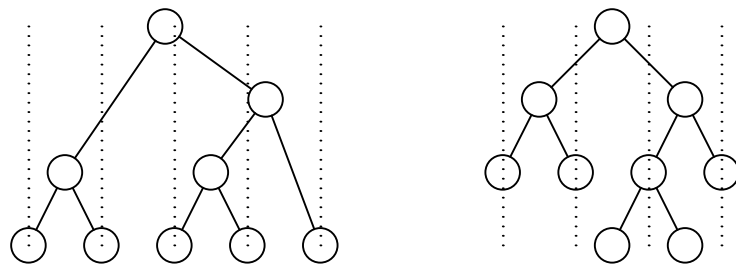
- Requirements tend to use higher-level concepts than implementation (programmer) view points.
- The higher-level concepts can be very different from lower-level ones.

Environment/User-view (input/output oriented)	Machine/Programmer-view (more detailed and algorithmic)
text-fi le = alternating sequence of words and word- separators end of sequence (in a word)	<ul style="list-style-type: none"> • text-fi le = sequence of word-characters and non-word characters • end-of-fi le indicating end of a word or non-word chars • opening/closing a fi le • reading one word at a time and skipping non-word characters or reading one character at a time detecting start/end of words
wordCount charCount equals the sum of word-lengths	<ul style="list-style-type: none"> • wordCount initialization and updates • charCount initialization; direct updates on reading word characters (need not recognize the equality condition)

MORE ON REQUIREMENTS

Two Softwares for Displaying a Binary-Tree:

- The softwares produce the following displays for the same test input. (The dotted lines, which are equally spaced, are added here to show the differences in relative node positions.)



Question:

- ? What are some differences that you notice between the two displays?
- ? Could we have avoided the surprise at the demonstration time by explicitly specifying what we want (or don't want)?

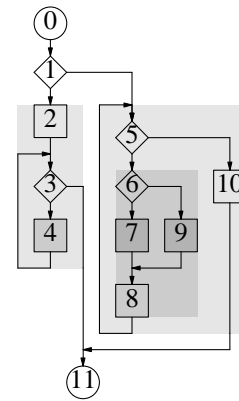
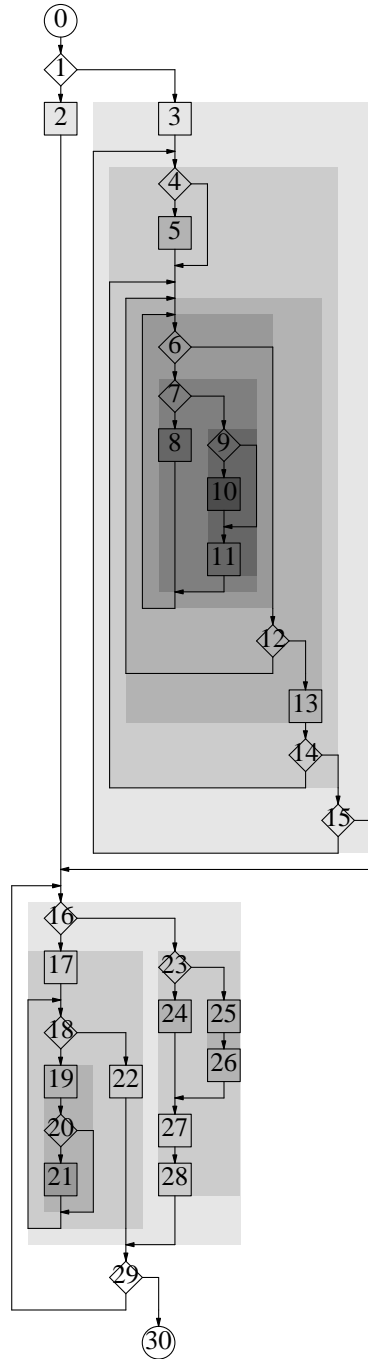
State your requirements clearly (unambiguously)
and without conflicts.

- ? Is there a good reason not to have given those clear specifications to the vendors?
- ? What are some possible differences in the algorithms for the two types of display?

Describing a thing in a way to distinguish it from others
may not be easy even when we see/feel the difference.

FIND SOME GOOD REQUIREMENTS FOR A FLOWCHART-DISPLAY SOFTWARE

- Use the following two example displays as a guide.



IN SEARCH OF REQUIREMENTS: DATA ORIENTED APPROACH

Identify Data (input/output) and Their Characteristics.

Entities and Attributes for Flowchart-Example:

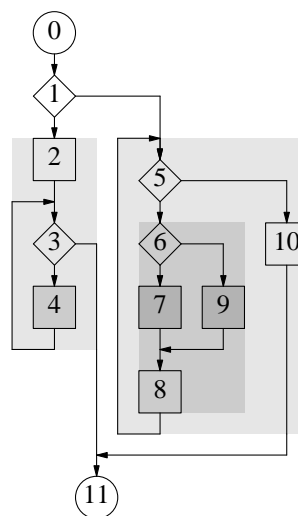
- *Nodes*: position, shape, and content, entry-position (for incoming lines), and exit-position (for outgoing lines).
- *ConnectingLines*: shape, start/end nodes.

Spatial Relationships (related to display operation):

- Spatial separation and relative positions of nodes and lines.

Other Entities and Relationships (also related to display operation):

- Block-structure among nodes; each block is a "super-node" with its own position and shape.
- Nesting relationship among blocks.



Question:

- ? Does this help to formulate the requirements?
- ? Find a better scheme for loop-back connection for do-while loops. Explain why the new scheme is better.

REQUIREMENTS FOR MENU-DESIGN

Menu: A list from which a user selects one or more items.

Example: Shown below is a two dimensional (multi-list) breakfast menu for a small restaurant. The person at the service counter marks the items selected by the customer.

<u>Main Items</u>	<u>Drinks</u>
<input type="radio"/> Scrambled Egg	<input type="radio"/> Coffee
<input type="radio"/> Sausage	<input type="radio"/> Coke
<input type="radio"/> Toast and Jelly	<input type="radio"/> Milk
<input type="radio"/> None	<input type="radio"/> None

Question:

- ? Is there a good reason for having the "None" item in each list?
- ? Why is coffee listed first in the drinks-section and why does the drink-section appear second (right)?
- ? How many different breakfasts can be ordered based on the above menu?

Evaluation Criteria of Menu-design:

- (1) *Exclusiveness* (disjointness of the list-items): can we choose more than one item in the same list?
- (2) *Completeness* (of each list): does it include all items offered. Also, are all lists present?
- (3) *Organization within a list* (ordering of items): does it make it easier for the customer to choose? (The designer may want to direct the customer to choose particular items.)
- (4) *List-placement*: corresponds to the order in which the customer is likely to make decisions about different category of items?

EXERCISE

1. In what way the following arrangement of menu-items is superior/inferior to the one shown earlier?

Main Items	Drinks
Scrambled Egg <input type="radio"/>	<input type="radio"/> Coffee
Sausage <input type="radio"/>	<input type="radio"/> Coke
Toast and Jelly <input type="radio"/>	<input type="radio"/> Milk
None <input type="radio"/>	<input type="radio"/> None

2. How do you provide a small/large option for each menu item?
3. Which of the two forms for submission of students' end-of-semester grade by course instructors is better and why? The instructor only fills in one of the circles for each student who is not marked *w* or *a*. (The forms comes with the Course-name, Semester, etc already filled-in and not shown here.)

Student Name	Fill-in one of the circles marked <i>A</i> , <i>B</i> , <i>C</i> , <i>D</i> , and <i>F</i>	withdraw or audit
...	<input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> F	
...	<input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> F	
...		W
...		A

Student Name	Fill-in empty circles with one of <i>A</i> , <i>B</i> , <i>C</i> , <i>D</i> , and <i>F</i>
...	<input type="radio"/>
...	<input type="radio"/>
...	W
...	A

4. The proposal submission web-pages for NSF (National Science Foundation) has a menu of three items as shown below for controlling access to the proposal by the Sponsored Research Office (SRO) of LSU. The person preparing the proposal selects one of circles for the SRO access control to the proposal.
- Allow SRO to only *view* the proposal but not submit
 - Allow SRO to *view and edit* the proposal but not submit
 - Allow SRO to *view, edit and submit* the proposal

What problems do you see with this menu in terms of the evaluation criteria (1)-(3)? Show a better menu design for the access control (in regard to viewing, editing, and submitting).

5. Consider a more general situation than Problem 4, where one gives controlled access to others (one person at a time) for editing and viewing a document that has clearly identifiable segments like "summary", "body", "financial data or budget", "figures", "tables", "special text", etc. Design a menu such that the menu is both easily extendible (e.g., new category of identifiable document segments may be added) and flexible in terms of the level of access allowed for each document segment.
6. If we want a 5th grader to choose two even numbers from a list of the numbers $\{4, 6, 12\} \cup \{1, 7, 17, 19\}$, what would be a good way and a bad way to present these numbers?
7. Shown on the next page is a sample display of the performance data for a particular student in a particular course. Design a display-control menu in which the student can indicate what he wants to see in the display. Note that there will be some mandatory items that must be included in the display as per the requirements of the University, although this can vary from course to course. In the example given here, the student's score and total score for each exam and homework fall in this category. An

instructor may on the other hand have options to turn off some other menu items so that a student cannot see the related information; for example "the rank in class" may fall in this category (and is not shown in the example display).

The design task here is to determine the contents of the menu and organizing them in a proper way so that the student can easily fill in the appropriate information based on his/her choice of items for display. Note that the actual display of the desired information will depend on the selected items in the control-menu.

A COURSE-PERFORMANCE DATA DISPLAY

Student: Sukhamay Kundu

Course: Software Engg, **Semester:** Fall **Year:** 2008

Grading scale: A=85-100, B=75-84, C=65-75, D=55-64, F=0-54

Performance-data for tests and homeworks: Current grade = B

Tests(90%)	#1(20%)	#2(20%)	#3(25%)	#4(25%)
My score (Total):	90(100)	30(50)
Class average:	87	40
Homeworks(10%)	#1	#2	#3	#4
My score (Total):	10(10)	10(15)	5(15)	...
Class average:	9	8	12	...

All test score (Total) = 120(150),	Weighted-value = 72(90%)
All homework scores (Total) = 25(40),	Weighted-value = 6(10%)
Total = 78(100%)	

Do you want to choose another course (y/n):

REQUIREMENTS FOR strcpy-FUNCTION

A Test-function for strcpy-function and Its Output:

```

#include <stdio.h>
#include <string.h>
#define LENGTH 10

void testStrcpy()
{
    int i;
    char str[LENGTH] = "abcdef";
    printf("str: '%s'\n", str);
    for (i=0; i<=LENGTH/2; i++) {
        strcpy (str, str+LENGTH/2-i);
        printf("i=%d, str: '%s'\n", i, str);
    }
}

```

str: 'abcdef'
 i=0, str: 'f'
 i=1, str: 'ef'
 i=2, str: 'def'
 i=3, str: 'f'
 i=4, str: ''
 i=5; str: ''

EXERCISE

1. Explain why the output looks this way, by showing the content of whole `str`-array after each iteration of for-loop. The operation `strcpy(str+3, str)` with `str = "abcdef"`, however, gives a run-time error (but no error if `str = "ab"`); explain why that is the case.
2. Use the UNIX command "man strcpy" to see the on-line manual page for `strcpy`. Then, formulate the requirements for a new function `safeStrcpy` as indicated below, including a proper choice of the return-value, which will avoid the kind of undesired behavior observed above for `strcpy(str+3, str)` and which destroyed the source-string.

```
int safeStrcpy(char *destination, char *source)
```

EXAMPLE OF REQUIREMENTS FOR AN AUTOMATED LIBRARY CHECK-OUT SYSTEM

Requirements related to a book:

- Searchable by author-name, book-title, and subject-keyword.
- Check availability, return-date, and hold-status.
- Check-out, renew, put/cancel hold, and return.
- Report lost, stolen, damaged/out-of-service.

Requirements related to renew operation:

- Can be renewed only by the current borrower.
- Can be renewed (on-line) only if the number of times renewed is below the renewal-limit.
- Can be renewed only if there is no hold.

Requirements related to Overdue/Early-return notice:

- Issued when the due-date is passed.
- Issued when there is an existing "hold" on the book.

Requirements for borrowing an on-hold book:

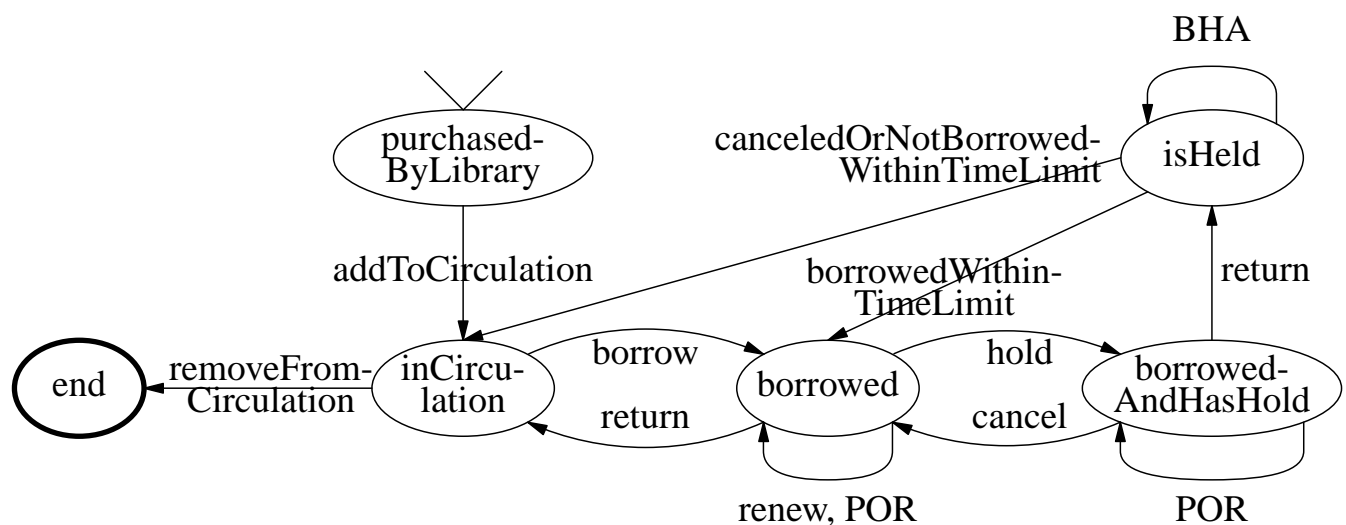
- Within a specified time-limit after the book is returned.

Relate each requirement with an entity or an operation.
--

FINITE STATE MODEL

FSM for a book:

- Assume for now that at most one hold on a borrowed book, and all books are returned (no lost-book).
- POR = Periodic overdue/early-return-required notice.
- BHA = Book-on-hold available notice.



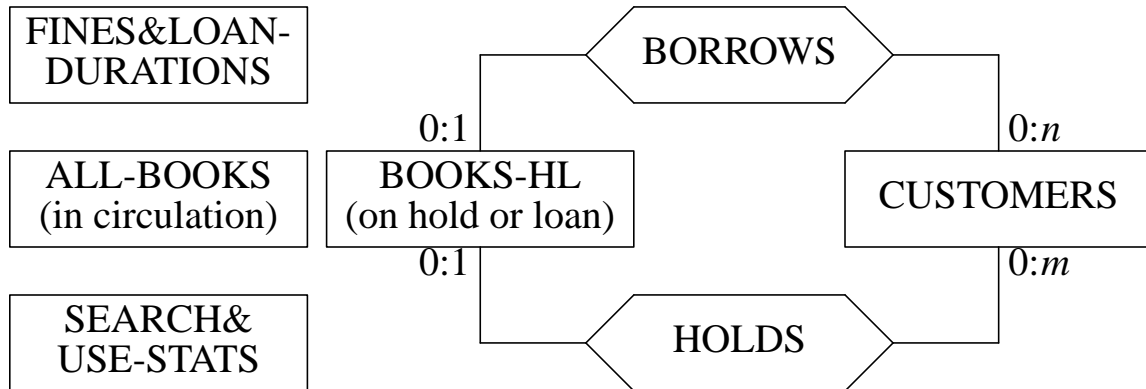
Example of Constraints:

- renew: (renewer.id = borrower.id) and (there is no hold)
- POR: POR-notice is sent only a fixed max. number of times.
- BHA: BHA-notice is sent only a fixed max number of times.

Question:

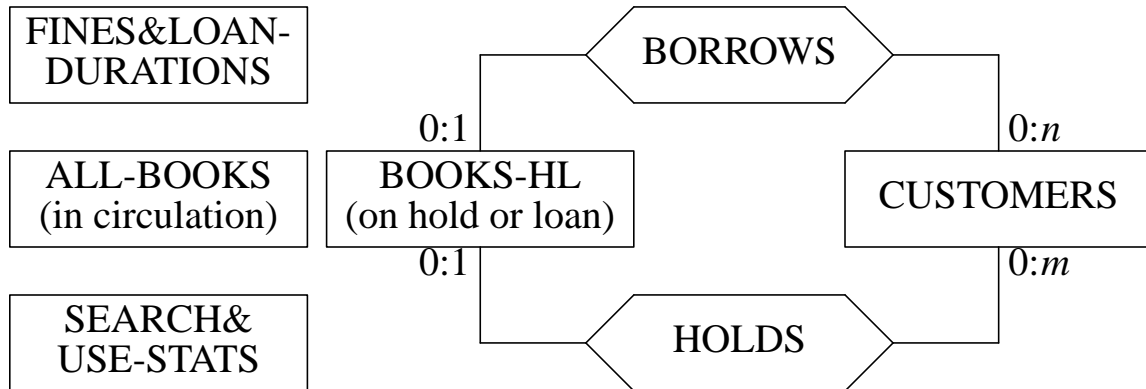
- ? Correct the above model so that one cannot renew the book after getting a POR-notice.
- ? Extend the model for the case of multiple-hold. (Hint: use the conditions $C_1 = \text{"Has 1 hold"}$ and $C_2 = \text{"Has } \geq 2 \text{ hold"}$; also indicate which of the constraints C_i is applicable for a transition, if any.)
- ? Show a finite-state model for a borrower; assume that at any time one can have at most 2 books borrowed and at most one hold placed.

ENTITY-RELATIONSHIP DATA-MODEL



- A customer may at any time have a minimum of 0 books and a maximum of n books borrowed.
- A book may at any time be borrowed by a minimum of 0 and a maximum of 1 customer.
- A customer may at any time have put hold on a minimum of 0 books and a maximum of m books
- A book may at any time have hold by a minimum of 0 and a maximum of 1 customer.
- A book in BOOKS-HL cannot have the two minimum cardinalities 0 at the same time.

ER-MODEL FOR AT MOST ONE HOLD(Contd.)



ALL-BOOKS:

(BookId, BookType, Title, Author, Publisher, PurchaseDate, PurchasePrice)

SEARCH & USE-STATISTICS:

(BookId, SearchCount, BorrowCount, TotalUseDuration)

FINES & LOAN-DURATIONS:

(BookType, CustomerType, LoanDuration, LoanRenewalDuration, HoldingPeriod, Fine, ReplacementCostPolicy)

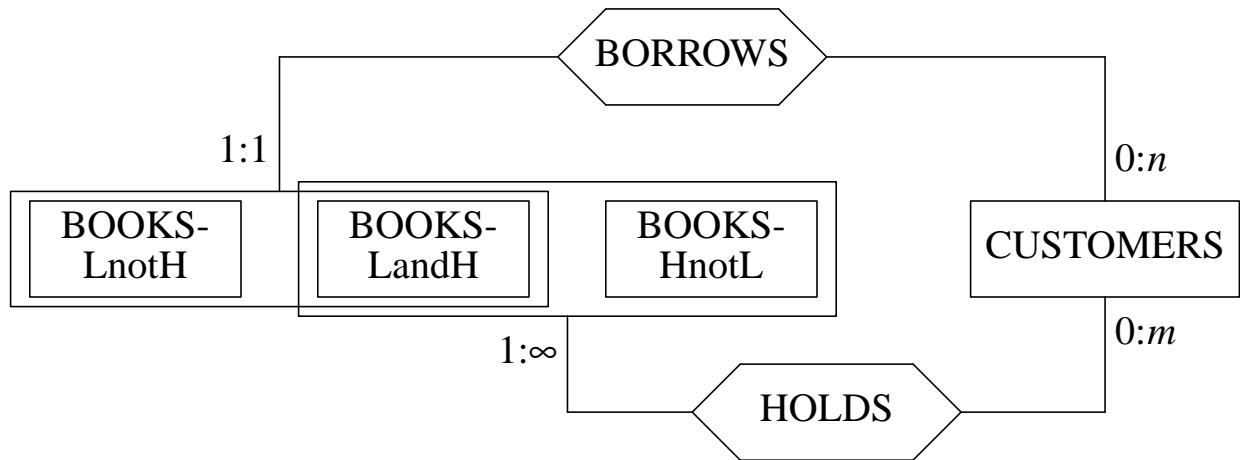
BOOKS-HL: (BookId, BookType, HasHold)

CUSTOMERS: (CustomerId, CustomerType, Address, TotalBorrowCount, TotalHoldCount, TotalLateRetCount, TotalLostBookCount)

BORROWS: (BookId, CustomerId, LoneDate, ReturnDate, DueDate, ReminderCount)

HOLDS: (BookId, CustomerId, HoldReqDate, HeldStartDate, HeldEndDate)

A REFINEMENT OF BOOKS-HL ENTITY



Not Shown In The Model:

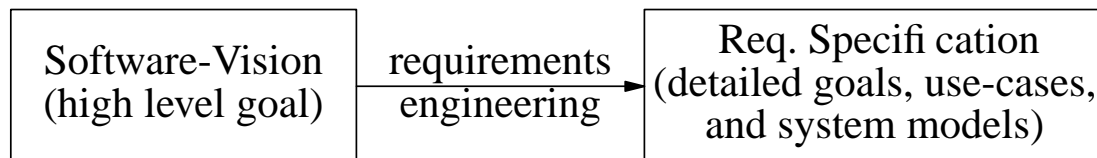
- For multiple holds, we need additional attribute *holdSequenceNum* in HOLDS-relation.
- The person currently borrowing a book cannot put a hold on it.

Note:

- The same person cannot put multiple hold on a book is captured by the key (*BookId*, *CustomerId*) of HOLDS.

WHAT IS REQUIREMENT ENGINEERING

- RE is a process that creates and maintains a system's requirements documents.



Question:

- ? What does the term "process" mean here?
- ? What is the final product of this process? What does it start with?
- ? If one says that he knows RE, is he saying that he knows "how" to carry out the process or is he saying that he knows "what" it is to produce?

Evaluation of Vision/Goal (feasibility study):

- Will it support business objectives (activities)?
- Will it work with other existing systems in terms of information exchange or will it be a stand-alone system?
- Will we buy existing software or develop (in-house or out-source)?
- Is there too much risk in acquiring/building the system?

CONTD.

Avoiding Missing Requirements:

- Group the requirements by user-types (a higher level classification) and use-cases. (Make a matrix of requirements vs. user-types/use-cases.)
- Group the requirements separately by output data-items and by input data-items.

Verification:

- Users verify the requirements to indicate that the requirements captures/states what they want the system to do.
- Analysts verify the requirements for consistency, completeness, realism, and verifi ability.

Validation

- Developers/testers validate (certify) that the requirements have been correctly implemented.

Verifi cation and Validation together imply that correct functionalities have been correctly implemented.

Management of Requirements Changes:

- If a requirement is relevant in more than one use-cases or more than user-types, then a change-impact needs to be analyzed with respect to all user-types and all use-cases.

EVOLUTION OF REQUIREMENTS

Two Ways of Simplifying A Requirement:

- (1) Simplify the functionality, an operation-centric approach.
- (2) Constrain/restrict the inputs, a data-centric approach (which can also simplify the functionality).

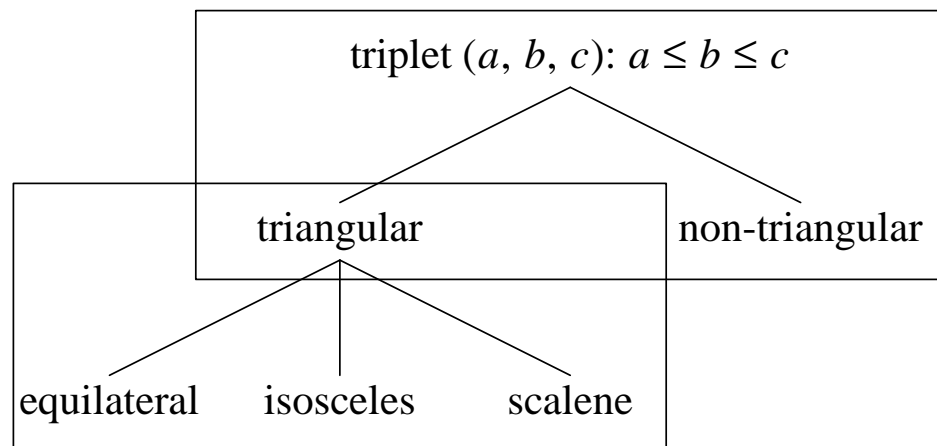
A Requirement and Its Two Simplified Forms:

Classify an integer triplet (a, b, c) , $a \leq b \leq c$, as triangular (equilateral, isosceles, or scalene) or non-triangular.

- (1) [constraining inputs] Classify a triangular triplet of integers $a \leq b \leq c$ as equilateral, isosceles, or scalene.
- (2) [simplifying functionality] Classify a triplet of integers $a \leq b \leq c$ as triangular or non-triangular.

Both Are Related to Data-Hierarchy Relationship Here:

Simplified functionality
(upper part of hierarchy)



Restricted input ($c < a + b$)
for the sub-hierarchy

BUILDING SOFTWARE FROM THOSE FOR SIMPLIFIED REQUIREMENTS

Pseudocodes for The Simplified Requirement:

```
function ClassifyTriangles(a, b, c) //a <= b <= c
{ print input lengths a, b, c;
  if (a == c)
    then print "form-an-equilateral-triangle";
  else if ((a == b) || (b == c))
    then print "form-an-isosceles-triangle";
  else print "form-a-scalene-triangle";
}

function IsTriangular(a, b, c) //a <= b <= c
{ print input lengths a, b, c;
  if (c < a + b)
    then print "form-a-triangle";
  else print "does-not-form-a-triangle";
}
```

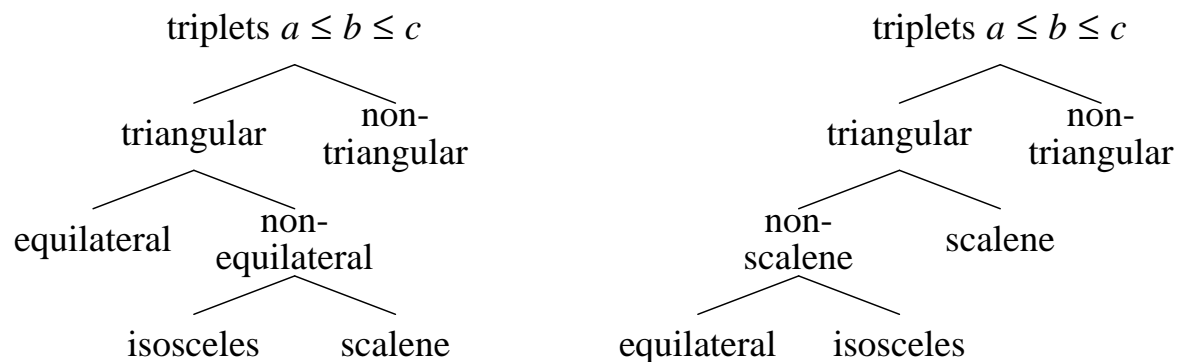
Pseudocode For Original Requirement:

```
function ClassifyTriangles(a, b, c) //a <= b <= c
{ print input lengths a, b, c;
  if (c < a + b )
    then //call old ClassifyTriangles
      if (a == c)
        then print "form-an-equilateral-triangle";
      else if ((a == b) || (b == c))
        then print "form-an-isosceles-triangle";
      else print "form-a-scalene-triangle";
  else print "does-not-form-a-triangle";
}
```

- (1) Simplify requirements to create early/initial versions of the software.
- (2) Complex requirement (software) can be seen as a result of evolutions of simpler requirements.

EXERCISE

1. Could we, in principle, further refine the triangle-hierarchy (see below) for viewing it as successive evolutions? Which of the two hierarchies below is better for this or are there some others (what are they) which are even better than both of these (and why)?



2. Is the logic in the following pseudocode correct for the original requirement? Does its structure relate to a simplified version of the requirement?

```
function ClassifyTriangles(a, b, c) //a <= b <= c
{ print input lengths a, b, c;
  if (a == c)
  then print "form-an-equilateral-triangle";
  else if ((b == c) || ((a == b) && (c < 2*a)))
  then print "form-an-isosceles-triangle";
  else if (c < a + b)
  then print "form-a-scalene-triangle";
  else print "does-not-form-a-triangle";
}
```

3. Show a new hierarchy where we are also to classify a triangle as one of {right-angle, obtuse-angle, acute-angle}.
4. Show how the notion of "evolution of requirement" applies to WordCharCounts-function shown earlier. Does it apply equally well for both forms of WordCharCounts-function (with and without restriction on word-lengths)?

REQUIREMENT EVOLUTION BASED ON HIERARCHY OF OPERATIONS

Hierarchy of Operations:

Consider the operations related to an entity (they will typically be connected with the relationships involving that entity).

- We say the operation op_1 is a sub-operation of op_2 if op_1 can be applied only after op_2 .

Example: Operations on a library book.

- Here, holdReturn is the return of a book with "hold", which is quite different from return of a book without a hold.
- Here, evolution means enlargement (addition) of functionalities instead of enhancement of existing functionalities (as was the case in triangle-classification).

A Possible Development Sequence:

- borrow \rightarrow return \rightarrow hold \rightarrow holdReturn \rightarrow not-return-lost \rightarrow search. (Return-operation can be tested properly only with borrow-operation in place.)

