

**AC 2007-3129: MOTIVATING WITH INQUIRY AROUSAL: CREATIVE
PROBLEM SOLVING IN ENGINEERING COURSES**

Mark Miles, United States Military Academy

C.M. Chewar, United States Military Academy

MOTIVATING WITH INQUIRY AROUSAL: CREATIVE PROBLEM SOLVING IN ENGINEERING COURSES

Mark Miles & C. M. Chewar
mark.miles@usma.edu; christa.chewar@usma.edu
United States Military Academy
Department of Electrical Engineering and Computer Science
West Point, NY 10996

Abstract: Many students pursue engineering disciplines because they enjoy creating systems and devices that are used to solve real problems. However, it is our belief that many engineering classes, especially at the introductory level, fail to engage students because they focus too much on objective data-manipulation exercises instead of encouraging inventive problem solving. This paper examines the effects of methods that allow students in an introductory level engineering course to use their creativity and inventiveness to solve problems, an approach referred to as inquiry arousal.

The approach used in this research was to modify a certain number of laboratory requirements to allow the student to apply an open-ended problem solving approach to obtain a solution. The goal of the assigned problems was to allow the student some flexibility to be creative while demonstrating fundamental mechanics. The authors obtained data and evaluated the effects of this approach through grade performance and student feedback. This paper attempts to measure the effects that assignments based on an inquiry arousal approach have on both the students' interest in the material and the students' level of learning.

Key words: Course Design, Problem Solving, Teaching, Creativity

1. Introduction:

As a student progresses throughout their academic curriculum, they will almost assuredly encounter subjects that they do not enjoy studying. Good students will overcome the frustration associated with working outside of their "comfort zone" and do their best to learn subjects that they do not enjoy. For many engineering students, learning mathematical theory or background, while important, is tedious; academic engagement is stimulated by an environment where they can apply some concepts to solve a problem.

This paper will discuss experiences with altering a course to include more opportunities for the students to solve open ended multi-concept problems. There is a great deal of published research that deals with the underlying philosophy and expected learning effects of providing "ill-formed" problems for student to solve [5]. This paper studies and offers recommendations on building creative problems into an introductory engineering course. Parts 2 and 3 of this paper describe an instructor's experiences and observations while altering a course. Part 4 of the paper presents data collected related to the changes in the course implementation. Part 5 discusses possible approaches to writing good questions, gleaned through our experience. The final part of the paper summarizes the key points.

2. Background:

One of the authors recently assumed course development responsibilities for an introductory course in Computer Engineering, a required course for both Electrical Engineering and Computer Science majors. The course was well designed and roughly followed the course text in how it presented material related to the subject. The plans for class instruction were reinforced through homework assignments and then tested on exams. Two multi-part projects (labs) involved applying various concepts learned in the class in a design fashion. It was a typical introductory course, in that the book presented problems which sometimes required designing and implementing a solution, but which all had a definitive right answer that the students could shoot for. The two labs were larger problems that involved more effort from the students, but also had an approved solution that was non-negotiable.

With the intention of concurrently redeveloping and teaching the course, we looked for improvements that could be made to invoke a deeper learning experience. For the first half of the semester, the Instructor/Developer maintained the idea that the out of class assignments were geared toward reinforcing the course learning objectives. He wrote challenging, multi-concept problems that had correct solutions the students could strive for. The lab and homework assignments were worth 55% of the overall grade, and since it was rare that any of the students failed to complete them, the grade distribution was a typical “bell curve.”

On the mid-term exam, the Instructor/Developer included questions that were meant to reinforce concepts the students had practiced on their problem sets. However, he was very disappointed in the results. His initial analysis was that they were “figuring out” the problems assigned, merely repeating the mechanisms of a solution method without really learning the underlying concepts.

2.1. Related Work:

Seeking to enhance our understanding about how we could engage these engineering students in a deeper level of learning, we turned to Driscoll’s discussion of *curiosity and interest* as a determinate of student motivation [1]. Cognitive theorists extend the general premises of motivational research, which explain human behaviors as resulting from specific decisions to fill some human need or *drive*, such as basic subsistence. Driscoll contends that when students decide how much to persistently engage in a learning task, they are weighing personal *values* (academic excellence) against opposing drive forces (the need to rest, participate in social events, or achieve in other activities). She suggests that the decision might be tipped in favor of the learning task if we leverage personal values, such as curiosity and opportunities to be creative, as sources of motivation.

Driscoll describes several approaches to harness student curiosity [1]. Some are simple, such as perceptual arousal with unexpected sensory events that can result from innovative instructional media and classroom methods. Driscoll also mentions how creating “fantasy” scenarios for students to participate in as they demonstrate learning serves to sustain curiosity for longer projects. Her final suggestion, which was based on the work of Keller [2], is called *inquiry arousal*. In this approach, a very complex and realistic problem is introduced for students to solve. The problem is designed to invoke “knowledge-seeking” behavior, where students ask for more information. Such behavior is then rewarded with additional clues or introductions to new problem solving techniques.

2.2. Our Approach:

We sought to enhance the level of engagement of the students in the Computer Engineering course by using an inquiry arousal approach. However, our efforts tailored this general approach somewhat, based on the nature of the course material. It seemed like the key questions to answer were:

- *How should we go about making the problem complex?*
- *What makes a Computer Engineering problem realistic?*

In struggling with exactly how to make a problem complex, we decided that an assignment should be loosely specified, so as not to suggest a particular solution methodology. To make the learning event more realistic in an engineering context, we decided to introduce the notion of different design teams competing for the distinction of creating the “best product.”

3. Trial 1:

With this approach in mind, we return our attention to enhancing the Computer Engineering course. This section describes a trial implementation of the inquire arousal approach that started mid-way through a semester.

```
(a)
origin 0x300
landscape:
    string "xxxxxxxxxx"
origin 0x310
    string "xSX  xx"
origin 0x320
    string "x xx x  x"
origin 0x330
    string "x  xx x"
origin 0x340
    string "x xxxx  x"
origin 0x350
    string "xxx  xxx"
origin 0x360
    string "x    x"
origin 0x370
    string "x xxxxxxxx"
origin 0x380
    string "x    D x"
origin 0x390
    string "xxxxxxxxxx"
origin 0x400
startpoint:
    word 0x0311 ;start coordinate
destination:
    word 0x0387 ;destination

(b)
checkleft:
    addi r1, -0x1 ;look left
    ld r4,[r1]
    addi r4, -0x58
    bz dontmoveleft
    addi r3,1 ;increment move counter
    call checkdone
    call movetank

dontmoveleft:
    addi r1, 0x1 ;reset pointer
    ret

checkdown:
    addi r1, 0x10
    ld r4,[r1] ;load character that is there
    addi r4, -0x58
    bz dontmovedown
    addi r3,1 ;increment move counter
    call checkdone
    call movetank

dontmovedown:
    addi r1, -0x10;reset pointer
    ret

checkdone: subx r4,r1,r2 ;see if current location and destination are equal
            bz done;
            ldi r4, 0x58 ;marks the location as having already been here
            st [r1],r4
            ret; return to caller if this is not true

done: ldi r6, destination ;this should be the end of memory
      addi r6,1 ;this should increment the pointer to one spot past the last
      st [r6],r3 ;this stores the number of moves your program took to comp
      hlt
```

Figure 1. (a) Syntax defining a maze, given to students as a possible input example that their solution should be able to handle. (b) Small portion of a student solution—a series of assembly language commands to handle the input conditions of the maze.

3.1. Assignment 1:

The first assignment after the mid-term exam was redeveloped entirely. The Instructor/Developer assigned a problem that involved the students having to use assembly code to help a robot navigate a maze (see Fig. 1). The problem was not worth more than any other problem set, but very little guidance was provided. A small amount (less than 5% of the assignment) of bonus points were offered for the most efficient solution.

As expected, the response that this particular problem invoked was “knowledge-seeking” behavior. Out of 22 students in the class, 14 came to the Instructor/Developer as they were working on the problem for feedback and guidance on how to take the next steps. Of those that came to the instructor for additional instruction, their attitude can be summarized:

“Now that I’ve done this and know where I’m trying to get to, I’m not sure what to do next.”

This provided the opportunity to provide clues, which were nudges toward analytical steps or inferences. Ultimately, the products that students submitted reflected many extremely complex and creative solutions to the problem.

3.2. Assignments 2 & 3:

Of course, we were left with uncertainty about whether the increase in motivation and effort observed on the assignment was due to the creative freedom inspired by the loosely defined problem or the competitive aspect of designing the best solution. Therefore, on the next problem set, the Instructor/Designer assigned a similarly open-ended question that asked them to design a memory structure. However, the competitive aspect was removed, and instead we sought to inspire creativity by explicitly rewarding efforts that produced higher quality solutions. Quality was measured in comparative terms of efficiency, since a more efficient memory system normally involves using memory components in a creative or inventive way. As part of the problem statement, a certain portion of the assignment grade (20%) was offered based on the efficiency of their solution. Again, the number of students that came for additional instruction was much higher than in the first semester and the quality of work, especially for the better performing students in the class, was well above what they had been producing in the first semester.

In the last lab given before the final exam the Instructor/Designer included another “inquiry arousal” portion worth 25% of the grade. This trial did not specify that inventiveness or creativity would be weighted any more or rewarded with any kind of bonus. The problem did not cause as many of the students to seek additional knowledge or interim feedback as the past two problem sets had, but did produce active discussion in the classroom. The work produced was again of an extremely high quality and showed some creative and inventive approaches not even considered when designing the problem.

3.3. Trial 1 Results:

The students performed much better on the final exam than on the mid-term. On the final exam, the level of difficulty of the problems was equal to that of the mid-term exam. The students achieved an average of 88% compared to the 79% average from the mid-term. During course end feedback, all the students were asked to rank the projects they had worked on throughout the semester, both in terms of difficulty and how much they think they learned. Though the results

showed a fairly equal distribution of how difficult the problems were, the students overwhelming stated that they learned more from the three “inquiry arousal” assignments. These results are included in Tables 1-4.

4. Trial 2:

The Trial 1 implementation provided encouragement that an inquiry arousal approach could be used successfully in engineering classes to generate motivation in learners. However, an offering of the same course in a new semester provided the opportunity to study the effects of the approach implemented throughout the course, rather than in just the second half.

4.1. Redevelopment to Support Inquiry Arousal:

The first half of the Computer Engineering course involves designing, building and evaluating a CPU based on the five stage execution model taught in most Computer Architecture textbooks [3,4]. In previous implementations of this course, students were given a lab with detailed instructions that led them to design a RISC-based processor with a specific instruction set. This lab reinforced very similar classroom instruction, and previously did not include any problems that encouraged the students to be creative in their solutions or approach.

In order to motivate student creativity, the Instructor/Designer acknowledged that course goals would have to be adjusted slightly and decided to incorporate more subjective goals in the first half of the course. The redevelopment effort to include an inquiry arousal element also resulted in an adjusted lab assignment. Rather than stepping through the RISC-based processor design in a hands-on lab, the design process was only taught in the classroom. Students were taught the basics of CPU design and walked through a detailed design of a datapath and an implementation of a control unit, the two key elements of a CPU. During quizzes they showed a fundamental grasp of “how” hardware components were designed to function appropriately. Pleased with the level of learning demonstrated by the students, the Instructor/Developer used the lab to challenge students to decide some of the capabilities the computer would possess, and then design that processor. Extra credit was offered for designing, building and testing CPU capabilities that extended beyond the basic design.

4.3. Trial 2 Results:

The students had a much more difficult time—and spent much more time—completing the lab than observed during the lab preceding Trial 1. Specifically, when the students were asked to design their own computer with the capabilities they desired, they had trouble working out the “why” portion of the design. They could not conceptually grasp how the elements of a computer datapath worked together or the effects small design changes would have on the rest of the machine.

However, the effects of having the students struggle through some of the design and engineering decisions had a positive impact on their mid-term test grades. The mid-term exam covered basic CPU design and testing and was of an equivalent level of difficulty from the previous semester. The students entered the course with a nearly identical class GPA compared to the students in the previous semester yet scored an average of 89% on the mid-term exam compared with the 79% average from before. There are other possible explanations for the grade discrepancy (luck, environmental conditions, different learning styles etc.), but the Instructor/Developer’s perception was that the students entered the test with a greater comfort

level with the material due to being forced to work through a lesser-constrained design problem prior to the test.

The second half of the semester was executed in a very similar manner to Trial 1. For the remainder of this Trial, students' attitudes, feedback and performance were very similar to the group from the previous semester.

5. General Results (Trial 1 and 2) and Student Feedback:

To help gauge the effectiveness of the inquiry arousal approach taken in Trial 1 and 2, students were asked to provide course feedback at the half way point of the course and again at the conclusion of the course. This section compares the results from Trial 1 and Trial 2.

The first table, Table 1, shows the impact on the major graded events. In comparing the trials, the students who had to work through labs with less structure (Trial 2) ultimately performed much better on the mid-term exam. Both groups of students had similar lab requirements in the second half of their semester and performed nearly identically on the final exam.

	Mid Term %	Final Exam %
Trial 1	79	88
Trial 2	89	86

Table 1: Overall grade performance on exams, by Trial

Table's 2 and 3 report an average from all the responses from the students to two questions on their course feedback. The students who took the course during Trial 2 found the first three labs harder, while they also claimed to learn more from the exercises. Again, there was almost no difference between how the students perceived the level of difficulty or the impact on their learning during the second half of the semester.

	Pre Mid-Term			Post Mid-Term		
	Lab1	Lab2	Lab3	Lab4	Lab5	Lab6
Trial 1	3	3.3	3.1	4	4.2	4.3
Trial 2	3.3	3.8	3.6	3.9	4	4.1

Q: How much did you learn from this Lab Assignment? (0-5, 0 being lowest)

Table 2: Student feedback from Mid-Course and End-Course Survey

	Pre Mid-Term			Post Mid-Term		
	Lab1	Lab2	Lab3	Lab4	Lab5	Lab6
Trial 1	3.8	3.6	3.6	3.9	4	3.8
Trial 2	4.2	4.5	4.2	4	3.9	4.1

Q: How difficult was this Lab Assignment? (0-5, 0 being lowest)

Table 3: Student feedback from Mid-Course & End-Course Survey

The final table (Table 4) also reports an average based on student feedback to the question of how much they thought the course “enhanced their ability as engineers.”

	Mid Term Survey	Final Survey
Trial 1	3.8	4.3
Trial 2	4.5	4.6

Q: I am learning concepts that enhance my ability as an engineer (0-5, 0 being lowest level of agreement)

Table 4: Student feedback from Mid-Course & End-Course Survey

The numbers indicate that the students during the Trial 2 believed that both halves of the course had roughly equal impact on their development as engineers, while the students from the first iteration weighed the second half of the course more heavily.

6. Trial 3:

Inspired by the success achieved with the inquiry arousal approach in the Computer Engineering course, we sought to examine its effects in a different course. We selected an Introduction to Programming course, a first course toward a Computer Science major or non-major “3 course” Computer Science Engineering Sequence. However, the 17 students observed in Trial 3 were all non-majors. The other author was an instructor in this course, and had project development responsibilities.

For the majority of the semester, students completed short projects in which they used several basic programming techniques learned across a multiple course lessons. To complete these projects, they had to decompose a larger problem into tasks that could be solved with the techniques that they knew. Then, they applied the technique to the problem sub-tasks and assembled the various parts. Most students had demonstrated mastery of the key concepts at this level of learning. However, the instructors were interested to see whether the course had resulted in students “learning how to learn” new programming skills. An inquiry arousal approach seemed perfect for this type of student assessment.

6.1. Inquiry Arousal Assignment:

A final project was designed using the inquiry arousal approach. The students were given a program for a game that worked, but with limited functionality. The code that they were given was not significantly longer than any other program they had already written, but it used a few concepts that had not been covered by the class. However, the textbook owned by all of the students introduced each of the new concepts in a similar manner to the concepts covered in the previous lessons.

The students’ assignment was to implement the remainder of the functionality. The list of features to be added was arranged so students knew the order they should be implemented. Implementing the most basic functionality extensions ensured a passing grade, while features further down the list (requiring more complex concepts) were necessary to earn higher grades. The task of implementing a basic feature usually only involved finding a similar feature that already worked, and adapting its code for the new feature. Therefore, students had to “inquire”

how the similar feature worked in the first place. More complex features required some experimentation, since there were no existing patterns to follow from similar features.

6.2. Trial 3 Results:

The 17 students observed generally performed well on the project, although the instructor was disappointed that few attempted the most difficult features. Most students, including some of the top performers, reported great difficulties with extending the project without fully understanding it. Students did not seem to be inspired or motivated by having to inquire why the code provided to them actually worked. While it was expected that students would come to their instructors for additional instruction with questions about the foreign concepts, few actually did. Instead, students seemed to attempt pattern matching wherever possible, which still trying to avoid thinking deeply about or looking up the new concepts.

One possible explanation for these disappointing results is that the project was the first of its type, yet introduced at the very end of the semester. Perhaps the students were just not willing to adapt to a new method. Introducing similar requirements earlier in the course might help in future trials. However, this general approach still felt quite exciting to use, and it is believed that with some refinement, it will serve as a useful instructional approach for this class as well.

7. Discussion--Creating Inquiry Arousal Problems:

This paper attempts to evaluate the impact on the student's motivation and the overall content comprehension caused by replacing objective "number crunching" problems with problems designed to invoke inquiry and creativity—complex, "open-ended" problems with realistic challenges built into the assignment. Throughout the description of the trials, we have already referenced some of the advantages noted after implementing inquiry arousal approach assignments, but there are also some possible drawbacks to consider before altering a course. We discuss those drawbacks, and then present recommendations for techniques to implement an inquiry arousal approach in an engineering course.

7.1. Potential Drawbacks:

The inquiry arousal approach calls for open-ended problems and ongoing interaction with students during a "knowledge seeking" phases of their solution process. Some of the disadvantages inherent in this include:

- Fairly assigning student credit is generally much harder
- Questions are often harder to write
- Providing clues during the "knowledge seeking" phase must be supported

Writing projects and establishing firm grading "cutsheets" is much easier when problems are purely objective in nature. A course likely has a list of course objectives that guide teaching, and an appropriate number-crunching problem tests the students' ability to succeed within that objective. Grading these types of problems is much less time consuming.

When you allow students some subjective "flexibility" in their solutions, you need to be able to still fairly assign credit for their effort and achievement. Does a student who designs and implements a complex solution, but makes a few small errors, deserve more credit than a student that goes for a simple solution and makes no errors? How much credit is awarded for a student

that uses course concepts taught in class correctly, but fails to solve the problem? These are just some of the issues that arise from trying to grade subjective problems and must be dealt with on a student-by-student basis. We found that grading was much more difficult and time consuming after switching to using open-ended problem types, but considered the extra effort worthwhile for the perceived benefit that it provided the students.

Another consideration must involve the increased interaction with students during their “knowledge seeking” phase. Since the approach is intended to make students think deeply about an ill-defined problem and “discover” the need to employ specific solution methodologies, they may need encouragement and nudging toward such solutions. Of course, this takes additional time and an offer of collaboration/additional instruction, which may not be possible for larger class sizes than the ones we used (maximum 22).

7.2. Recommended Techniques:

Writing assignments for inquiry arousal that still tested the course objectives was often difficult to do. We found that such assignments must provide the students with some guidance and “boundaries” within which to work, or they will not be able to solve or even understand the problem. Of course, how one ultimately writes an assignment depends on the type of concept being taught, ability of the students, and the response types being targeted. Here, we summarize a few considerations we thought important while writing questions that both allow for creative problem solving on the student’s part *and* still assess mastery of course objectives.

- **Subjectively evaluate a portion of the assignment.** After a few iterations of assignment writing, our goal with most problems on labs and homeworks was to have 25% of the credit of the problem be open to student creativity. This allowed a student who could just mechanically demonstrate concepts to achieve a passing grade, while still rewarding the students who could “apply” the concepts to solve various problems. It also was a significant enough portion of the grade to warrant extra effort being spent on developing good solutions.

An example of this approach was to add a “now go further” portion on to an already clearly defined problem. For example, there is a portion of one lab in which students needed to evaluate the design of a simple digital circuit and be able to make changes to it. The question was assigned as a three-part problem. The first two parts were very objective and asked them to implement and test certain functions (multiple solution approaches were possible, but there was one very well-defined goal). The final part was completely open ended and said “design, implement and explain another feature that would be useful to this device.” This last part was worth 25% of the grade and required the students to really think about fundamental circuit design principles and engineering tradeoffs. When the student is faced with the “how hard is this going to be to do vs. how much credit am I shooting for” dilemma, we try to tempt them with a chance to be creative.

- **Give minimum, measurable standards that can be met and easily evaluated by the students.** This portion of the assignment, or the basic version of an answer, *does not* require creativity or further inquiry. It is straight-forward and provides the confidence necessary to embark on the more challenging portion of the assignment. Students inherently know when they have successfully reached that level.

One lab in the Computer Engineering course involves designing memory systems for a computer. Past iterations of the course asked students to design and test a particular memory system that they discussed in class with well-defined parameters. In the spirit of inquiry arousal, the question was altered to read: “you have XX Bytes of memory available to you; design a memory system by allocating this to an instruction cache and data cache, then determine your hit rate.” Left at this, it could have been a nightmare assignment. However, the Instructor/Developer then calculated the worst “hit rate” (a measure of a cache’s effectiveness) a successful design could achieve with that much memory and published that hit rate as a minimum standard (the 75% solution). Any hit rate above the “C” level of credit would involve students using the memory they were given in creative and more complex ways. This gave the students an extreme amount of flexibility in the design, but still allowed the instructor to have a good measure of success.

- **Give the students multiple levels of solution sophistication to shoot for, let them decide exactly “where” to go for each level, but be ready to support the quest for “how”.** Provide examples of what an “A”, “B” and “C” solution could be, and ensure that each higher grade involves a less constrained problem. There should be multiple solution methodologies that can be used to get to more sophisticated solutions, and the assignment materials should only hint at which to use. Students will experience frustration coming up with the “how” for each successful level, so be ready to support with additional clues.

A portion of the Computer Engineering course involves learning how to program in Assembly Code. The major lab that goes with that block of instruction requires the students to write a program to control an object which reacts to its environment through sensors. A “C solution” may simply travel horizontally and detect obstacles, where a “B solution” may be able to go vertically and horizontally and navigate obstacles. However, an “A solution” may determine the most efficient way to navigate around multiple obstacles. In this particular problem, the simplest solution really just shows an ability to write and implement a program in assembly code, while anything higher shows the ability to think creatively and use assembly code in more complex ways. Once a student achieves a “B” level solution on their own, they may be able to see that adding a path optimization algorithm could get them to the next level, although they have no idea how. Encourage the student at this point to come for additional instruction, and give them a rough idea of what to do next.

- **Encourage competition.** The most successful inquiry arousal assignments contained the element of one solution being designated as the “best” solution. “Best” can be determined by measuring either by some objective criteria (time, hit-rate, lines of code, etc.) or instructor subjective judgment (most unique, most complex, most creative, etc.). Being designated as “best” should result in recognition and/or extra credit.

This technique had the effect of challenging the top students in the class and pushed them to go beyond the content taught in the classroom setting. Since the best students set their own learning goals, the technique has the added benefit of keeping the material manageable for the students that have a harder time learning the concepts.

8. Conclusions:

Though this paper does not offer scientific evidence to prove that any approach to problem writing is better than another, it does review our experiences and results in using an approach to inquiry arousal and creativity-inspiring assignments in engineering classes. Certain types of test or project questions can impact many learning factors, such as student interest, motivation, and confidence level, but they are very difficult to measure.

Our effort started with a course that was designed around objective problems—sufficiently testing the material, but offering no flexibility and inspiring no creativity on the part of the students. Seeking to engage students on a deeper level, and motivate them to mastery new concepts, we turned to a goal of inquiry arousal, and implemented several variations of the general approach. We also probed the use of this approach for testing to see how well we, as instructors, have taught students *how* to learn. We observed the effects of altering a certain percentage of such graded events, using student feedback, personal observations, and grade performance on course events as indicators of the effects alterations caused.

Based on our observations, we offered insight toward the drawbacks, and we presented recommendations for implementing an engineering course inquiry arousal approach. Through discussion of several assignments in three trials and two different courses, we offer specific experiences that shaped our broad understanding about the characteristics of effective curiosity and creativity-inducing assignments. As this is an initial effort, there are many directions for future work. Comparing our specific approaches to other related work will be essential as we form the next new questions. Employing more rigorous data collection methods and evaluating this data with well-established analytical techniques will support future scientific conclusions. It will also be interesting to continue adapting student assessment instruments in other engineering courses to include inquiry arousal intentions. Broader experience with this approach will certainly reveal many other implementation considerations.

Ultimately, we are encouraged by the experiences reported in this paper, which lead us to believe that students work harder at solving a problem and take more pride in their solutions when the product of their work is unique and personal.

References:

- [1] Driscoll, Marcy P. *Psychology of Learning for Instruction*, Allyn & Bacon, 2000.
- [2] Keller, J. M. "Motivational Design in Instruction" In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models*. Erlbaum, 1983.
- [3] Parhami, Behrooz. *Computer Architecture: From Microprocessors to Supercomputers*. Oxford, 2005.
- [4] Patterson, D, and Hennessy, J. *Computer Organization and Design: The Hardware/Software Interface* (second edition). Morgan Kaufmann, 1998.
- [5] "Thoughts on problem solving." Available at: <http://www.engin.umich.edu/~problemsolving/open/blooms/>