ENHANCING CRITICAL THINKING SKILLS IN FIRST YEAR ENGINEERING STUDENTS

Norman Lawrence[1], **Giles Thomas**[1] **and Denis Visentin**[2]
[1]Australian Maritime College, Australia
[2]University of Tasmania, Australia

ABSTRACT

Students come to first year engineering degree courses with a variety of learning styles and experiences. Many students can solve problems but lack the ability to critically analyse data and results, and are unable to present the data in meaningful ways. There can be an expectation that these generic attributes are either already present, or will be developed by osmosis through the degree program. This paper describes how learning outcomes are achieved, generic student attribute skills are fostered and experimental reporting skills are nurtured in a specialised unit in experimental methods.

Students are introduced to programming, problem solving and analytical skills using software such as LabVIEW and MATLAB. These skills will then form a basis for later studies. The focus is on developing the 'tools' for problem solving early in the degree program, and giving generic skills a prime focus rather than allowing them to become a peripheral issue to the 'content' of the engineering degree.

Students are encouraged to explore the relationship between theoretical predictions and experimental results. In particular students are encouraged to critically appraise experimental data collected, apply error analysis, discuss and present results meaningfully and reach conclusions.

INTRODUCTION

The Department of Maritime Engineering at the Australian Maritime College provides four-year, full-time Bachelor of Engineering Degrees in:

- Naval Architecture,
- Ocean Engineering,
- Marine and Offshore Systems.

Student numbers vary between 70 and 85 in the first year engineering degree program.

The focus of this paper is on the content of the unit 'Experimental Methods' and its delivery to a multi-disciplined group of first year engineering students. A description is provided on how learning outcomes are achieved, generic student attribute skills are fostered and experimental reporting skills are nurtured.

UNIT PHILOSOPHY

Studies by Felder and Brent(1) confirm that students come to first year engineering degree courses with a variety of learning styles and experiences. Often students can solve problems but lack the ability to critically analyse data and results, and are unable to present the data in meaningful ways. Students can experience a significant contrast between school where they may rely more on being told how to do things and University where they are required to be independent learners who can work things out for themselves.

Sometimes at university there can be an expectation that these generic attributes are already present. Bertolina and Thompson(2) assert that in many current engineering courses, skills such as teamwork, critical analysis, leadership and lifelong learning have often been developed in an osmotic manner. We believe that it was important to do something beyond expecting engineering undergraduate students to improve as critical thinkers as an emergent property of their degree studies. Critical thinking requires the ability to make and defend conclusions based on evidence and we felt that this was best done within a defined context (in this case experimentation). There needs to be a clearly stated intent to develop critical thinkers. This can be more easily achieved if there is a mapping of the 'skill' through the degree. For instance, the use of Excel might be introduced in a first year mechanics unit (as part of an assignment requirement) reinforced and extended in an electrical unit, and then becomes expected knowledge in later years, and required in units.

However, we were reluctant to develop a new unit where only critical thinking would be taught. Some courses in analysis or perhaps numerical methods take this approach, where the techniques can be separated from the content in the same way that a student might learn calculus, and only later apply it to mechanics or fluid mechanics after assimilating the knowledge. The problem with this method is that the students often fail to see the relevance of the skills they are learning (as has been the case with many mathematics and programming units in the past).

It is difficult to *teach* critical thinking. There must be willingness on the student part to engage the ideas. There is also a need for a setting for the ideas to be developed. Too trivial a problem will not be engaged; too difficult a problem will not allow the student to become involved at the right level. In Experimental Methods the unit has little explicit content beyond the methods, techniques, tools and ideas presented, there are a number of practical sessions, where real data is recorded, real measurement equipment is used and real safety standards must be met. While the cognitive content of these laboratories is not essential to the unit, it is important that the students engage real problems, rather than applying critical thinking to an artificial problem, or something they have seen many times before. The practical sessions are designed to introduce the students to new and diverse experimental situations such as a towing tank (used to physically model a ship's hydrodynamic behaviour), engines, fluid flow, and electrical equipment, in different laboratory environ-ments. The sessions also have a variety of different measurement techniques and instruments – Venturi tubes, multimeter and oscilloscope, tachometer, mechanical loads, data acquisition using temperature sensors – to measure a diverse range of physical variables. The generic concepts in the unit can then be seen applied to the particular problem at hand. The students are encouraged not merely to make the appropriate measurements, but to consider the factors affecting the measurements, how the values might be estimated, and how to check results as they go using 'back of the envelope' calculations.

Engineers Australia(3) list ten generic attributes considered desirable for graduate students to possess and at AMC these ten attributes have been mapped to individual units in our degree programs. The generic attributes addressed in Experimental Methods are as follows:

1. The ability to apply knowledge of basic engineering and science fundamentals;
2. The ability to communicate effectively;
3. The ability to understand problem identification, formulation and solution, and
4. The expectation of the need to undertake lifelong learning and capacity to do so.

In this unit we are seeking to equip students with a generic 'problem solving toolbox.' The toolbox can be considered to consist of real tools and conceptual tools.

Real tools are used to help students solve a problem. For instance a calculator helps students to solve problems, but they need to know how to use it first. Experimental Methods introduces students to new problem solving tools and shows them how they can be used to solve problems in general. Examples of real tools include:

- MATLAB, LabVIEW, Excel,
- Data analysis, error analysis, statistics,
- Sensors and transducers.

Conceptual tools provide students with new ways of thinking, and so allow the students to analyse totally new problems. Examples of conceptual tools include:

- Algorithms and flow charts, numerical techniques,
- Research skills, experimental design, report writing,
- Analysis techniques, and mathematical modelling.

Clearly there is a measure of overlap between the real and the conceptual tools.

The work described here is underpinned by the belief that critical thinking and lifelong learning can be fostered by providing a rich learning environment where students are encouraged to read, discuss and use the concepts in a variety of challenging projects.

IMPLEMENTATION

Experimental Methods is a 12.5% credit unit taught in first semester to first year students, with the following content:

- Computer Programming Skills (26 hours, 1 lecture and I tutorial per week, plus laboratory visits as required)
- Experimental Skills (26 hours, 1 lecture and I tutorial per week)

Each of the above dot points will be discussed more fully. Whilst separate topic headings have been identified for the delivery of this unit, it should be remembered that the acquiring, developing and assessing of individual skill sets is often integrated across the unit.

Computer Programming Skills

We do not seek to teach the students to be programmers, rather we are teaching them the rudiments of programming in the context of MATLAB and LabVIEW. There is less need for the modern engineer to be able to program in a language such as FORTRAN or C++. It is however important that they can use languages such as MATLAB and LabVIEW to solve problems, to manipulate and present data, and to simulate processes. The majority of the students had either never programmed at all, or had only encountered Java. Hence it is assumed that students have zero programming skills before the course begins.

Students are first introduced to LabVIEW over a six-week and then to MATLAB over a further six-week period. Both software packages use the format of one lecture and one tutorial per week.

Some generic programming skills are also introduced. Algorithm design is presented. Students are expected to generate ‘solutions’ to problems by producing a flowchart or algorithm of the problem. It is intended in 2006 that students will bring their ‘solution’ to the tutorial session as a flowchart (for LabVIEW) or a pseudo-code algorithm (either is acceptable for MATLAB). Too often students think the computer will solve the problem for them, and start typing before thinking about the problem. The concept of solving a problem takes a new form when a computer is used. The same is true of experiment design, an experiment is designed to solve or investigate a problem, and it is the design of the experiment which is one part of the ‘solution’ and the mechanics of performing the experiment is analogous to ‘implementing’ the solution. This also maximises the amount that can be performed in a tutorial session, since implementing a solution and debugging is often a time consuming exercise for new programmers. After the first session introducing the software environment of LabVIEW and MATLAB, the next 5 sessions present the concepts of functions, loops and control structures, symbolic expressions, matrix representation and I/O, and numerical techniques in terms of real engineering problems.

Students are encouraged to think of LabVIEW not only in term of data acquisition or instrument control, but also for general purpose applications such as database development, data analysis programs and network communications.

Particular emphasis is placed on developing student’s ability to approach computer programming in a logical manner. The need to create flow charts, develop algorithms before writing code is stressed. Importance is placed on developing good error checking techniques within the code, and for debugging purposes. The graphical nature of LabVIEW means that programming, using a flow chart as a basis, is a relatively easy concept for students to acquire. LabVIEW is an easier way to approach the harder concepts of loops and case statements.

Students undertake a major individual assignment using LabVIEW in which they are required to write a computer program for data analysis and presentation. Using data, they have acquired themselves in a laboratory session in the towing tank, students write a program to determine the full scale resistance characteristics of the test vessel.

The 6 weeks spent on MATLAB (one lecture and one tutorial session per week) introduces the possibilities of MATLAB, and encourages students to use the software as a generic problem solving tool. It is intended that students will be expected to use LabVIEW and MATLAB in future years of the degree program, and hence build on their skills acquired in year 1.

Experimental Skills

Students receive 13 formal lectures on basic experimentation theory, including:

- Occupational Health and Safety in Laboratories.
- Instrumentation and measurement.
- Error analysis and statistics.
- Data analysis and report writing.

Understanding and application of the theory is underpinned with weekly tutorial sessions and laboratory work. For example, students are introduced to error analysis concepts using Excel through a combination of lectures and tutorials.

In 2006 the tutorial problems will be completely linked to the real problems students have encountered in the laboratory sessions. So they will model fluid flow, perform a linear regression on their loaded beam data, plot the loading effects of electrical meters using user interface and loops.

Students should enter the second semester of their first year of studies with a toolbox of generic problem solving skills associated with relevant software.

Sepahpour(4) believes that Laboratory experimentation is a critical link for a thorough understanding of scientific and engineering theories. He states that development of the laboratory components plays a significant role in the enhancement and completeness of engineering courses. In Experimental Methods, students are also encouraged to develop research skills by recording information about the experiment, not merely the relevant raw data. Before they leave the session, a simple graph or evaluation is performed so that in a one hour session they have been introduced to a laboratory environment, equipment and measurement devices, physical concepts, and generated a result. These results may then be further analysed in later tutorial sessions.

Laboratory work in a curriculum is very resource intensive but it provides significant benefits such as: the acquiring of skills in using real equipment; gaining an insight into phenomena of interest; developing measurement skills, obtaining an appreciation of experimental errors and an awareness of any simplifying assumptions that have been made. Laboratory exercises can also be used to develop in the student the skill of analysing experimental measurements and reaching relevant conclusions.

Feisel and Rosa(5) state that, 'When undergraduate students go to the laboratory, it is generally not to extract some data necessary for design, to evaluate a new device, or to discover a new addition to our knowledge of the world. Each of these functions involves determining something that no one else knows or at least that is not generally available. Students, on the other hand, go to an instructional laboratory to learn something that practicing engineers are assumed to already know. That something needs to be better defined through carefully designed learning objectives if the considerable effort devoted to laboratories is to produce a concomitant benefit.'

In this unit, students are required to submit a single formal laboratory report on one experiment. Only when students have completed all the experiments are they allocated an experiment to write up. Students must keep complete records of all experiments. The report forms a significant part of the coursework assessment (20%). If we want to teach and assess report writing skills, how do we ensure that students are learning the right skills, and not just making a single failed attempt, from which they will obtain only minimal feedback? We attempted to help students 'get it right the first time' by allowing them to see the assessment criteria. They know how many marks are allocated for what and hence know the relative importance of each section (it is amazing how quickly student adapt to the selection pressure of *marks awarded*). They also see a breakdown of the important elements, which they pay much more attention to than in an assignment specification. For instance, it is difficult to get first year students to see the importance of correct referencing. It is easier to say 'there is 5% for correct referencing, this is the easiest 5% you will ever get, and learning it once will get you 5% for every assignment from now until the end of your degree. Here is how to do it right . . .' This applies to other oversights by undergraduates that examiners find continually, such as presentation of graphs (title, axes labelled, units, etc) and figures (labelled and numbered, referred to in the text).

Another method utilised for the laboratory report was to have two submissions for a single report. A draft report due date was held, when the report was *peer-marked*. Students were given a marking scheme with detailed criteria. We used tutorial sessions for students (about 16 per group) to analyse, mark, provide feedback and for the lecturer to make general comments. After reading the report, each section of the marking scheme was examined, with the lecturer giving general comments (using a specific paper, chosen from a willing student) on the expectations of that section. Students were asked to give feedback comments as well as a mark. At the end of the session, students were asked to consider the mark they had been given for each section, the reasons for the mark and to consider improvements. Most of the assessments were reasonable and disputed elements (for example referencing styles graphing techniques) gave an opportunity for the lecturer to address some of the areas about which students were unsure of the requirements.

This formative assessment was designed to serve two purposes: to give the students two bites at the cherry, so that they were made aware of the deficiencies in their initial attempt, and also to take on the role of marker, and hence see the task from a different perspective. Students were noticeably more critical in their analysis of other peoples work, and then could see the inherent problems in their own. Last year this method had some good feedback, but too few actually submitted the draft report. In 2006, there is a mark associated with submission of the draft report, if the draft is not submitted; the maximum final mark is 80%. Hopefully this will make the exercise more valuable, by ensuring that all students submit a draft report.

OUTCOMES

While critical thinking may be difficult to foster in undergraduates, it is even more difficult to evaluate. Essentially students can be evaluated against a number of different criteria. We can see how well the students can use each element of the toolbox. This can be performed in a simplistic manner by testing the student's skills in each independently, or seeing how the toolbox is used to attack a particular problem. The students at the end of the LabVIEW and MATLAB sessions demonstrated competence in each of these areas. But are they aware that these tools can now be used for many different problems that they will see in their later studies? First year students have been found to be using LabVIEW and MATLAB to check assignment problems, and to generate graphs with numerical analysis for laboratory reports for other subjects. Clearly some students had added MATLAB to their generic problem solving toolbox. The evidence that some students are using the software taught in Experimental Methods in other units, where its use is not expected, is encouraging. But the best way to ensure that the programming elements continue to be used is to reinforce their use throughout the degree program.

Another important criteria is how the student's themselves see the knowledge they have acquired. Students evaluated the unit, and its delivery through a questionnaire. The responses were generally positive (there were no responses below the neutral point) and some responses in particular were encouraging. In particular students commented that they:

- Found the unit to be intellectually challenging
- Learned something valuable;
- Enjoyed participating in class discussions and being encouraged to ask questions.

Performance on assignments and exams, where the tasks and questions were designed to assess critical thinking, had mixed results. The report writing exercise was performed well, with students demonstrating the ability to research a topic, present their own ideas and critically evaluate results. The exam showed some deficiencies in critical thinking, which may have been due to the pressure of the examination situation. It was obvious in the lead up to the exam (and at times in the unit in general), that some students were confronted by the non-standard delivery of the unit: a typical comment was '*just tell me what to do so I can do it*'. Some students appeared not to engage the thinking aspect of the course, continuing to focus solely on the *content*. For instance, an examination question where the students were asked to present an algorithm for a programming solution proved difficult, with many offering either complete code or spurious calculations.

In general the interaction in tutorial sessions was very satisfying, with students willing to engage new ideas and propose solutions to different problems. Not only was the assimilation of the ideas demonstrated, but it was clear that many students were thinking creatively about solving problems.

CONCLUSION

In the newly introduced unit, Experimental Methods, we have sought to be creators of a culture of critical thinking among students rather than just presenters of information. The unit uses a number of different presentation styles and learning environments. Weekly tutorial sessions combine short laboratory exercises, library sessions, computer lab programming and spreadsheet exercises, and traditional problem solving tutorials. There are also different environments within this structure, using a variety of laboratory equipment and settings (in an attempt to go beyond the 'staged' first year experiment), as well as a number of software environments. We hope that this exposes the students to the environments of a modern engineer, where not only the laboratory is important but the computer as well, and indeed the interface between computer and experimental apparatus (for instance, the computer as a data acquisition and control device).

The subject is used to foster generic student attribute skills and shape students ability to collect data from a variety of sources and critically analyse that data. We believe that providing an early focus on critical thinking in a context that is meaningful, will allow students to progress with more confidence, and approach new areas of study and environments with the tools for success. However, further work needs to undertaken to establish the effectiveness of embedding these 'tools' in students regular toolboxes.

REFERENCES

1. **Felder, R. M. and Brent, R.**, 2005, '*Understanding Student Differences*', Journal of Engineering Education, Jan 2005, pp 57-72.
2. **Bertoli, A. and Thompson, L.**, 2002, '*Integrating Graduate Capabilities into Engineering Programs*', Proceedings of the 13th Australasian Association for Engineering Education, Canberra, pp 127-132.
3. **Manual for the Accreditation of Professional Engineering Programs**, The Institution of Engineers, Australia, 1999.
4. **Sepahpour, B.**, 2003, *Design of an Affordable Model laboratory for mechanical and Civil Engineering programs*, Proceedings of the American Society for Engineering Education Annual Conference. Session 1526.
5. **Feisel, L. D. and Rosa, A. J.**, 2005, *The Role of the Laboratory in Undergraduate Engineering Education*, Journal of Engineering Education, Jan 2005,121-130.