ENGINEERING STUDENTS THROW AWAY THEIR CALCULATORS

Dr. D. B. Lowe, Dr. C. A. Scott and Mr. Ravin Bagia
Faculty of Engineering
University of Technology, Sydney
PO Box 123 Broadway NSW 2007
Ph: +61 2 9514 2397   Fax: +61 2 9514 2435
e-mail: {dbl, cascott, ravin}@eng.uts.edu.au

SUMMARY

Informatics is a first year subject in the new Bachelor of Engineering/Graduate Diploma in Engineering Practice degree. All engineering students must do this subject irrespective of their intended engineering discipline. The focus of this subject is to introduce students to computational and information tools (such as the Internet, spreadsheets, and programming languages) which support engineering, as well as looking at the effective and professional use of these tools. Informatics has a strong thread running through it whereby students are expected to discuss and contemplate a wide variety of issues related to the computational tools they are using. Examples of the issues covered include: privacy; accuracy of content; fraud; security; the professional and ethical responsibility of engineers for the results of their calculations (bringing up issues such as the need to “know” the answer before using any computational tool); selecting the best tool for the job; and limitations of computational tools. This paper will review the activities and challenges the students faced, describe the various issues we raised, the expected and actual outcomes arising from the activities, and our plans for the future of the subject.

INTRODUCTION

In a remarkable turnaround, 1st year engineering students at the University of Technology, Sydney are being encouraged to spurn technology and throw away their calculators. The students are taking responsibility for their calculations and are using their minds before their calculators. This has resulted in more than one headache and quite a few surprised expressions when students realise what their brains can achieve.

In 1997, the Faculty of Engineering of the University of Technology, Sydney undertook a major overhaul of its undergraduate Engineering programs. This included redeveloping the structure, content, and focus of all undergraduate subjects. The essence of this redevelopment is a common core for all undergraduate engineering degrees, with the students focusing the work through choices of field-of-practice majors and electives. One of the subjects forming the core of the new degrees is Informatics, a first year engineering subject which must be completed by all undergraduates irrespective of their chosen Engineering discipline.

In developing the core subjects for the new degree, and in particular the development of Informatics, it was recognised that the existing degree programs (and most existing engineering programs) suffer from several significant problems. These include:

- Most courses develop specific skills in various tools but they fail to recognise that the range of tools which students will encounter in engineering practice goes way beyond what can be even partially addressed within a formal degree program. Indeed, many tools (and the technologies which underlie them) are changing so rapidly that there will be little similarity between those available at the start of the students degree and those at the end of the degree. This means that the students will regularly encounter tools with which they have no experience or familiarity. It also means that the students often lack an ability to determine the most appropriate tools for given problems.

- Related to the above point is that the students often lack the ability (or confidence) to continually develop their own skills. As part of their professional engineering practice they will be constantly required to develop not only their knowledge, but also their skills in utilising the tools which support them in their practice. Within a fairly short time from graduation, they are also required to mentor other colleagues in the use of these tools. This means that the students not only need to learn specific tools, but need also to “learn how to learn” so that they may continually refresh their existing tool skills and acquire new ones.
Although the use of tools is regularly taught in the context of engineering *problems*, it is often not taught in the context of engineering *practice*. For example, software development or programming subjects will often teach programming as applied to engineering problems (e.g. how to write software for engineering analysis) but will rarely consider issues such as the *impact* of errors or inaccuracies in programs, and the professional responsibilities with respect to the developed software.

Given the above observations, the purpose of this subject is to assist the students in developing an understanding of the types of tools and resources which allow them to manage the information, models, complex calculations, and processes associated with carrying out engineering activities. We shall begin this paper by looking at the subject’s educational objectives in a little more detail. We will then look at the subject structure and content which we have developed to address these objectives, and some of the early outcomes which we have observed.

**SUBJECT PURPOSE AND EDUCATIONAL OBJECTIVES**

The development of *Informatics* was undertaken using a very methodical approach. We began by considering how the subject could contribute to the overall course objectives, and then refined these contributions into a set of specific educational objectives for the subject. Let us begin by considering the overall course objectives.

**Graduate Attributes**

In redeveloping the undergraduate B.E. courses at UTS a set of desirable graduate attributes was defined [1]. These graduate attributes provide a definitive description of the characteristics which we believe are important for our graduates. The attributes also act as threads providing cohesion between subjects. In developing the structure and content for *Informatics* we began by looking at these graduate attributes within the context of the overall focus of the subject and determining which attributes the subject could best contribute to. The following partial extracts from the development documentation for *Informatics* to provide some examples of how we perceived *Informatics* should contribute to some of the graduate attributes:

**Professional Formation:**

*Attribute 1. Values and social contexts*

_Critical awareness of codes of ethics, their role and limitations:* Students will develop an awareness of their ethical responsibilities with respect to interpretation and utilisation of computational results.

*Attribute 2. Management skills*

_Experience in applying systems thinking where appropriate and approaching complex problems:* *Informatics* will develop an understanding of the ways in which different types of tools can aid in managing the complexity of engineering problems.

*Attribute 3. Technical expertise*

_Recognition and application of fundamental mathematical concepts and methods:* *Informatics* will develop an understanding of categories of tools aimed at supporting fundamental mathematical concepts and methods. In particular *Informatics* will develop an understanding of how tools aid in applying mathematics, the relationship between the mathematical concepts and the tools which support them, the capabilities and limitations of these tools, when it is appropriate to utilise these tools.

_Recognition and application of fundamental limits and principles of physical sciences:* *Informatics* will develop an understanding of the relationships between the fundamental limits and principles of physical sciences, and the limits and principles of tools which support the engineering utilisation of these principles. *Informatics* should also develop an understanding of concepts of accuracy and errors.

**Personal Development:**

*Attribute 1. Maturity*

_Critical and independent thinking:* *Informatics* will promote the ongoing development of critical and independent thinking through critical analysis of informatics tools and resources, their relevance, limitations, etc.

_Awareness of the importance of self-motivation and taking responsibility of one’s own decisions:* *Informatics* will promote the ongoing development of an understanding of the responsibility for appropriate utilisation of tools and resources in supporting decisions.

**Academic Development:**

*Attribute 2: Information literacy*

_Experience in seeking, accessing, retrieving, and evaluating references from a variety of sources:* *Informatics* will promote the ongoing development of an understanding of the types of tools, and the usage of these tools, which support identification and utilisation of information from a variety of sources.

*Attribute 3: Problem posing and solving*

_Familiarity with the process of conceptualisation and articulation of problems in terms of systems and processes:* *Informatics* will develop the students ability to utilise informatics tools in the conceptualisation, visualisation and articulation of engineering problems.
Recognition of relevant principles and useful approaches for identifying and structuring a problem: Informatics will develop the students understanding of how informatics tools can be used in supporting the identification and structuring of problems.

Awareness of problems as constructs based on values, interests, and perceptions of needs: Informatics will develop the students awareness of the relationships between the structure of problems and how these problems are expressed within tools which support reasoning about, or manipulation of these problems.

Familiarity with a variety of problem solving strategies: Informatics will develop the students understanding of different classes of tools which support different strategies to problem solving (indeed, according to the brief, this is a core focus of Informatics).

Awareness of the need and familiarity with ways of validating solutions: Informatics will develop the students understanding of the limitations of informatics tools, and the need to validate solutions (and mechanisms for performing these validations).

Subject Educational Objectives

The above contributions to graduate attributes were analysed to determine the specific educational objectives of the Informatics subject. One significant observation is that there are three apparent levels in terms of the contributions. First is the targeted development of understanding and skills related to specific areas (such as skills in using a particular programming language or a particular type of spreadsheet). The subject aims to develop specific skills for several reasons. Developing these skills will be important in developing the students’ confidence with respect to the use and role of tools. They will also provide the students with the skills required to support their learning in other areas (such as using modelling tools in systems design and spreadsheets for analysis of laboratory results). Finally, it will help the students understand the broader issues discussed elsewhere in the subject.

Second is the development of general understanding and/or knowledge related to the area of informatics tools and resources (such as understanding the different classes of computational tools which are available). We explicitly do not want to only develop skills in using particular tools, but rather to develop the students ability to understand general classes of tools and to have the ability to confidently approach the development of their own skills in using new tools. The third level of contribution is students’ understanding related to the professional application of their skills. A major goal of the subject is to ensure that the students develop an understanding of the implication of their professional activities. For example, if the students develop some software which controls a robot which subsequently malfunctions and injures or kills someone, are they responsible?

Based on the above contributions to the development of graduate attributes we identified a set of subject educational objectives. Again, these objectives are too numerous to include here, but the following partial extracts provides appropriate examples:

Objective 1) Informatics aims to specifically develop a deep understanding of:
(a) the types of engineering problems which can benefit from the use of additional resources and tools, and what these benefits are, especially in terms of managing the complexity of engineering problems
(b) categories of resources and tools which are available to support the carrying out of different aspects of engineering activities, the breadth of these resources and tools, and their strengths and limitations
(c) which types of tools are most appropriate for given types of problems and specialisations, and the need to validate solutions from these tools
(h) how informatics tools and resources relate to the culture of engineering
(l) how engineering tools relate to, and support the management of, reliability and safety.

Objective 2) Informatics aims to develop specific understanding and skills in:
(a) utilising the tools which assist engineers in identifying and utilising information available within libraries, on the Internet, and other suitable sources
(e) utilising the tools which assist engineers in resource management
(f) utilising the tools which assist engineers in risk management and decision making
(g) utilising the tools which support different stages of the design process

Objective 3) Informatics aims to continue to develop and refine the students understanding and maturity with respect to:
(a) critical and independent thinking, especially with respect to critical analysis of informatics tools and resources and their relevance and limitations.
(b) understanding of the responsibility of engineers for appropriate and ethical utilisation of tools and resources in supporting decisions and carrying out engineering activities.
(c) issues of sustainability.

SUBJECT STRUCTURE

Using the subject learning objectives as a starting point a structure was developed. Prior to considering this structure, we considered a number of constraints which impacted on the subject structure.
Subject Constraints

The first major element which was important to take into account was the huge variation in the level of understanding and skills in the commencing students. Although it is normal to expect variations in the prior learning of students, especially in first year subjects, this is particularly noticeable in subjects which focus at least partly on technical and computing skills. We have found that it is common to find variations ranging from students with little or no exposure to computers (though this is, admittedly, becoming more rare) to students who have very extensive (albeit rather narrow) experience. Recognition of Prior Learning (RPL) needs to be taken into account in the way the subject is run. The problem is further complicated, however, with students perceiving the ability to use a computer package as a true understanding of the purpose and uses of that tool.

A second element which needed to be considered in designing the subject structure was the impact on resources. The subject will typically have up to 320 students enrolled every semester. To accommodate this many students in laboratories at one time is impractical. Similarly, anything which can be done to ease the burden on staff or facilities (such as tutorial rooms) needed to be considered.

Structure

Based on the above observations, a flexible structure was designed for the subject. This structure is based around a series of subject modules. These modules focus on different aspects of the subject, and are broken into two main streams: skills and issues. The issues modules cover the development of understanding of classes of informatics tools, uses and limitations of these tools, the relationships between these tools and meta-skills in developing skills in using tools. The skills modules cover specific skills in different tools, including spreadsheets, the Internet, operating systems, programming, and a number of other tools.

Students are required to complete a set of compulsory modules, and one or more elective modules. To accommodate the huge variability in student prior learning and experience, the students are given the option of negotiating to replace compulsory modules (in which they can demonstrate appropriate prior understanding) with extra elective modules. In this way we are encouraging students to utilise the opportunity to extend themselves and maximise their learning as much as possible.

The various modules includes large classes (lectures involving presentation and explanation of material), small classes (tutorials and workshops), individual study (tutor-supported study involving one-on-one interaction between a tutor and a student), and independent study (study undertaken solely by the student) supported by self-learning modules. Each of these modes of learning/teaching has different resource implications, but provide an overall balanced approach to the subject.

To provide a sense of cohesion to the subject (especially in light of its module-based nature) during the first week of the subject, students are allocated to a specific tutor. This tutor takes the students for all small class sessions, including both issues and skills modules. This tutor becomes responsible for the students overall subject program (including aspects such as negotiating with them exemptions on compulsory modules and maintaining their marks for the subject) and will normally be the students first point of call in the case of questions or problems related to subject logistics.

Assessment

The assessment in the subject is designed to provide a strong focus on formative assessment rather than summative assessment. With the described subject structure it is impractical for every module undertaken by a student to have individual assessment tasks, as this would result in an excessive number of assessment tasks within the subject. Nevertheless it is important that the extent of the students understanding of both skills and issues be effectively evaluated. To achieve this the assessment in the subject is split into two components: Mastery assessment and Advanced assessment.

All students are required to complete the Mastery assessment component, which solely determines the students’ pass/fail grade but not their final mark. If the student passes the Mastery component then they are guaranteed a minimum mark of 50-Pass, even if they do not attempt the advanced Assessment components. If they fail any component of the Mastery assessment then they fail the subject. To satisfy the Mastery requirements, the students must satisfactorily complete all the subject modules (including any negotiated elective modules), and
pass the final oral/practical exam. Satisfactory completion of each module is based on evaluating, on a Pass/Fail basis, the student achieving an acceptable level of understanding. This is determined through the module assessment tasks and through the examination of a Journal that the students maintain throughout the subject. The Journal is an important element of the assessment and the students are encouraged to record their reflections as they engage with the subject matter. This approach removes the common problem of “mark-accumulation” and focuses the student on developing an appropriate level of understanding.

Those students who wish to earn a grade beyond a Pass can attempt the advanced assessment component. The advanced assessment component (of which every element is optional) determines the student’s grade once they have passed the Mastery criteria. Note that irrespective of performance in the Advanced assessment component, students cannot pass the subject without passing the Mastery assessment component. The Advanced assessment components involve a major programming assignment, and a learning contract which requires the student to negotiate a task which demonstrates understanding of two or more of the non-programming skills modules and at least one major aspect of the issues modules.

The Mastery/Advanced approach to assessment has been used extensively in similar subjects (notably software development subjects) over the last few years and has proved to be particularly effective [2]. This structure encourages students to focus on their learning, as distinct from focusing on mark accumulation, yet provides the scope for more ambitious students to obtain a grade which reflects their understanding.

EXAMPLES

Introduction to Programming Module: Algorithm Design

Informatics is different to many introductory programming subjects in that it offers students a choice of programming language. To avoid duplication a generic Introduction to Programming Module was developed. The focus of this module was not programming but rather problem solving. Experiences gained from other programming subjects has indicated that those students who struggle often have weak problem solving skills. Programming was introduced as a mechanism for implementing a solution to a problem rather than allowing students to work under the misconception that programming is a problem solving tool of itself. The lecture component of the module introduced the problem solving strategies, the strengths and weaknesses of computer programs, and mechanisms for dealing with complexity. The lectures were concluded with some examples to ensure that the students did not view the module as representing the only approach to software development and that there is much to learn beyond Informatics.

The key to the module was a set of tutorial/workshops involving hands-on problem solving. Students are asked to bring a deck of playing cards to these workshops. The aim is to first learn to solve problems, then to formalise the algorithm, to recognise that a given algorithm may be optimal for one problem and not necessarily for a similar problem, and finally recognise which types of algorithms are suitable for implementation using a computer program. Some of the key benefits gained by students, as indicated by their journal reflections, are the acknowledgment that there is more than one solution, and that they will not be able to find all solutions to a problem and a team approach may be more appropriate. Examples of problems used to develop problem solving skills include finding a given card in the deck, sorting a deck of cards, and scoring a Blackjack hand. These examples also introduce the uncertainty in user specifications with students raising questions such as “[When sorting] is the Ace below the 2 or above the King?” . Student feedback from these lectures and workshops was very positive. Some students even found the sessions entertaining with some tutors demonstrating a few card tricks.

Issues Module: Professional Responsibilities

One of the aspects considered in the Issues module is the students’ understanding of their ethical, moral, and professional obligations, as an engineer, in the appropriate use of information and tools. This includes aspects such as safety and reliability, sustainability, and critical evaluation and effective communication of the results obtained from tools. One of the ways in which this is achieved is through the use of several case studies which revolve around situations raising a number of difficult issues.

The first of these utilises a hypothetical situation described in “The Case of the Killer Robot” [3]. This book describes a situation where a software engineer is indicted for manslaughter based on his involvement in the development of software which controls a robot. The robot has malfunctioned and killed one of its operators.
We required the students to read various material relating to this hypothetical case and then reflect on various issues which are raised; the reflection guided by a series of questions such as

1. Develop a list of criteria which you believe could be used to decide whether a programmer should be held accountable in a case such as the Killer Robot.
2. Should Randy Samuels be indicted for manslaughter?
3. If you were pressured to release a product (such as the Killer Robot) before it was safe, reliable, or effective what would you do?
4. Write your reflections on how being a member of a development team could influence your approach and your sense of professional responsibility!

Following these reflections we ran a workshop for the students in which we focused on developing the students understanding of their ethical, moral, and professional obligations (as an engineer) in the appropriate use of information and tools. We then gave the students another set of readings related to a true story where a software design fault in a Therac-25 medical radiation machine resulted in the deaths of 7 people [4]. Again, the students are asked to read certain material and then document their reflections based on a set of guiding questions.

**OUTCOMES**

At the time of writing the subject has just completed its first offering. Feedback on the subject has been obtained formal and anecdotal evaluations. Several important observations can be made. First, the students have shown a high degree of enthusiasm for the subject. In particular the technical nature of the much of the subject gains the attraction of the students, and despite their initial reservations they become quite involved in the issues aspects of the subject. A number of the subject tutors have been approached by students outside normal class times, and become involved in animated discussions regarding ethical and professional questions.

Second, student feedback indicates that the overall structure of the subject is appreciated since it offers considerable choice in both what is learnt and how it is assessed. On the whole the Journal is seen both as a means to document reflections and as a tool for venting minor frustrations through the semester. Students have also made positive suggestions regarding some alterations that may be made to the emphasis given to different components of the subject. Feedback from staff involved in teaching the subject during the semester has been used to re-visit the level of assessment in some of the components in order to provide a better balance throughout.

It is worthwhile returning to the opening paragraph

… 1st year engineering students … are being encouraged to spurn technology and throw away their calculators. The students are taking responsibility for their calculations and are using their minds before their calculators. This has resulted in more than one headache and quite a few surprised expressions when students realise what their brains can achieve.

There is much anecdotal and some formal evidence indicating that the students are developing, very early in the degree, a sense of the context which defines how they can (and should) approach the practice of engineering in a professional contexts.

**CONCLUSIONS**

*Informatics* has been presented so far only for a single semester and further improvements to the subject are being developed. Anecdotal and formal feedback from students is very encouraging. The students have shown that they are willing to accept the challenge that we set them of thinking and reflecting on the subject material rather than accepting it simply because an “expert” presents it. They have been willing to engage in lively discussion regarding the broader issues related to practice of engineering in a societal context. An implication of this outcome is that the staff involved in teaching this subject must be equally willing to engage in such discussions. Creative ways of presenting the issues and the material bring the subject alive. For most of us involved in the subject, the experience has been a rewarding one.
ACKNOWLEDGEMENTS

The authors wish to acknowledge the invaluable contributions of other staff involved in the development and implementation of Informatics: James Boswell, Martin Evans, Ron Meegoda, Austin Mack, Richard Lai, Ben Rodanski and Ali Saleh.

REFERENCES