Get with the Program: Integrated Project
Instead of a Comprehensive Final Exam in a First Programming Course

Beverly K. Jaeger, Richard Whalen, Susan F. Freeman
College of Engineering, Northeastern University

Abstract

Part of our responsibilities as engineering educators is to continually revise and update our curriculum, including assessment methodologies and procedures. We need to ensure that our selected evaluation methods are not only fair and challenging to the students, but also relevant to the requirements of their future employers and to the demands of technological advances in engineering. In this paper, we describe the motivation and set out the strategies for changing a component of our computer programming assessment of first-year engineers from a written final exam format to an integrated programming project. This project was administered at the end of the semester across several sections of a first-year problem solving course that uses and ultimately integrates C++ and MATLAB to introduce students to engineering analysis and design. The details of the rationale, project development, assignment types, and final outcomes are set forth in the paper. The intent of this discussion is to provide a working model that other educators may follow to assist in developing an effective and representative programming assessment method that can be used by multiple instructors. To that end we will also make our work available to other engineering educators for their own adoption.

Introduction

At Northeastern University the first-year engineering curriculum is common for all majors, and there are typically 14-16 separate sections of approximately 30 students each. For the past decade, our introductory computer programming courses in engineering, like those of many other universities, had administered a hand-written final exam to comprise a substantive portion of the students’ overall grade. This evaluation method was supplemented by quizzes and computer-based assignments throughout the term. When initially electing to use the exam option, several primary factors and concerns were considered. The first was resources; at any given time, there had been only 64 computers available within the program, precluding the possibility of testing on computers simultaneously. Other considerations in developing a final exam were the constrained time frame that the students would have to complete their work at the end of the term, the freshman population’s relative inexperience at programming, maintaining consistent standards across multiple sections, and establishing a common final assessment process among several instructors. Further, we also wanted to limit opportunities for students to engage in unauthorized collaboration. In the past, administering a common final exam seemed to best satisfy these concerns and requirements.
**Final Exam Format.** The final exam that had been previously administered was two hours long, open book, open notes, no shared material, calculators permitted. The format consisted of short answers, calculations, evaluation of coding excerpts, and the requirement to generate a flow chart and hand-written code to solve one or two word problems. It was necessary to compose detailed questions, with short-response solutions, such as “What does this code return?” or “Write the call statement for the function in the program below” in order to cover as many aspects of the course as possible and still allow time to score the exams and make final grade computations. At times, the final exam questions had to be so specific that the students were locked into a relatively constrained solution path. This compromised any element of uniqueness in their work and differentiation in their skill levels that may have existed. Therefore, their performance and grades were not truly individual and perhaps not representative. The written final exam tested fewer components of the course, and lacked adequate performance detail because of time limitations and the constraints of hand-written code. The further drawback naturally was that the grading instructor could only theoretically assess the answers and could not test for computational adequacy, as the programs clearly could not be executed.

**Authentic Assessment.** The additional disadvantage of a written exam format was that it did not adequately typify or exemplify how programming is best learned, applied, and evaluated and was at best a confirmation of a student’s overall quiz performance. We realized that it would also be more beneficial to require students to address more substantial and realistic problems for the final assessment, yet a written final exam cannot and did not adequately provide for this. These factors are better represented by more computational and functional work methods and require authentic assessment methods, which more closely mirror the expectations of the professional world. The exam also did not adequately allow the students time to plan their answers, and there was no defined element of presentation of their final solutions.

In the mid 1990s there was a concerted movement in education to research, identify, and incorporate effective methods of authentic assessment. This effort was supported and funded by the US Department of Education (Kerka, 1995). The results were as expected: Authentic assessment methods that incorporate a wide variety of techniques designed to correspond as closely as possible to ‘real world’ student experiences are most effective and representative (Custer 1994; Rudner and Boston 1994). Further research has shown that projects and work based on authentic assessment have longer-lasting influences on students’ confidence and marketability (Borthwick, 1995; Stemmer, Brown, & Smith, 1992).

In the College of Education at Illinois State University (ISU), the faculty support for authentic assessment has resulted in classes on how to develop such an evaluation instrument. Professor Jabraun at ISU states in the preface of one such lesson, “Some educators feel that paper and pencil tests miss the essence of what students should be learning. Those educators –and you may be among them– would like to replace, or at least supplement, paper and pencil tests with tests that call for ‘real world’ skills. This assignment is intended to give you practice in developing such an instrument.” These introductory statements are followed by exercises to help faculty develop authentic assessment instruments for their courses.
It should be noted that authentic assessments must be carefully designed and evaluation criteria rigorously selected for the process to be successful in practice. Some of the characteristics of effective authentic assessments—which were incorporated into the integrated project at Northeastern— are as follows (Custer 1994; Rudner and Boston 1994):

- Engaging, meaningful, worthy problems matching content and outcomes of instruction
- Real-life and occupation-based applicability
- Multistaged-demonstrations of knowing, knowing why, and knowing how
- Emphasis on product and process, conveying that development and achievement matter
- Rich, multidimensional, varied formats, both on-demand and cumulative
- Opportunities for learner self-evaluation and self-correction
- Cognitive complexity-requiring higher order thinking skills
- Clear, concise, and openly communicated standards
- Fairness in scoring procedures and their application

To convey the process of integrating authentic assessment into our first-year programming course, the sections to follow will describe: (1) the curriculum in which the subject course is embedded and the course objectives, (2) the plan for efficiently and effectively developing sets of integrated projects with a faculty team along with the process for identifying common topics for each of the projects, (3) formats, instructions, insights, and techniques for administering the integrated project sets simultaneously across multiple sections, (4) quantitative data from faculty and students using time and grade records from the integrated projects, and (5) qualitative student and faculty feedback on the project. The latter two sections in particular support the theory that students are now provided with an opportunity to apply and present their knowledge in a representative way – and that they generally succeed in doing so. Finally, we review both the benefits and drawbacks of this endeavor and make suggestions for strategies that engineering educators may use when adopting similar means of assessment.

The Curriculum and Course

*The Engineering Program.* Northeastern University is distinguished by the cooperative education program it offers its students; accordingly, the College of Engineering strives to offer its students more than a theoretical education. There is a strong emphasis on practical application of technical knowledge and the development of analytical problem-solving and decision-making skills. Also, like many universities, we expect the students to develop and demonstrate advanced proficiency in both verbal and written presentation. In teaching the “Engineering Problem Solving with Computation” course, the focus is not solely or even primarily on the students’ knowledge of the logical components of the relevant software. Rather, the course is centered on the practicality and applicability of logical solutions to problems in real life using the available software tools. It is this notion of functional knowledge and its communication that the student engineers acquire as their most valuable “commodity” to take with them to their professional working lives. Therefore our focus must be placed on this competency when determining appropriate evaluation methods for this course. In looking at this point of focus, the integrated
project was developed to carry out this task, thus stepping away from the formal examination process. It is also the project’s requirement of a business letter or proposal at the conclusion of the programming work that further sets the project apart from an exam and emulates expectations found in the corporate sector in terms of professional correspondence.

The Course of Interest. Our current first-year programming course, Engineering Problem Solving with Computation, now covers two forms of programming software, C++ and MATLAB, to teach and apply the course components. This unification requires that these software tools must now be taught to approximately 400 students, using 64 computers over one 15-week semester. In combining these problem-solving tools, we faced the challenge of developing a final evaluation procedure that tested both the logical and computational components programming application. Highlighted the unique benefits derived from each, simultaneously addressed the problem-solving elements of the course, and possessed characteristics of real-world engineering challenges from a variety of engineering disciplines. We not only sought an assessment process that better addressed the primary factors and concerns we had identified, but one that could also be developed and administered by several instructors in a time span that was not prohibitive within the demands of the university environment in terms of faculty coordination and time investment.

We reviewed options for other methods of assessment and determined that using programming projects that integrated both types of software applications was the best alternative. We have found that the integrated project sufficiently addressed our concerns and at the same time successfully presented the students’ knowledge of the intricacies of the software tools. After developing and implementing the integrated project, we now believe this is not only a better alternative, it is currently the best final assessment method for our first-year programming courses. With the structure of the course using the two complementary programs, we can also teach the students to solve problems across software platforms, to demonstrate how two programming applications communicate with each other, and to learn how to select the “best tool” for the job according to the features of each software language. Writing code on paper during a defined time period, as occurs during an examination, was and is no longer able to provide sufficient evidence of the students’ level of mastery of the intricacies and thought processes required to navigate logically through the planning, execution, and analysis of complex computations.

Accordingly, our College of Engineering, in line with other Universities, has now replaced the final examination component of this course with the jointly-developed integrated project sets assigned at the end of the term. Numerous projects have been developed and are derived from several engineering-related disciplines. Within each section of a programming course, multiple projects have been assigned. Each professor assigns 2-4 different projects in each class and each project requires the student to utilize both software programs in solving one substantive real-world problem. Each student undertakes a single project over a 1½ week period. Students are commissioned to formulate a strategy, produce code, execute their code on a computer, and analyze and report their results. Their work can then be properly evaluated for its programming adequacy, accuracy, functionality, and applicability. Each student’s project and its solution draw upon all elements of the course, requiring problem formulation, code planning and development, a final business analysis, and a written presentation of the proposed solution.
**Other Evaluation Methods:** In addition to the other forms of assessment that are used throughout the semester, the integrated project is only one element of the students’ course grades. As had previously been undertaken when the final exams were used, quizzes are still administered nearly every week on recent material to ensure that students have individually mastered the concepts before moving on to the next set of topics. The students readily become accustomed to these, and most report that the quizzes compel them to keep up with the course content and reading. Homework assignments are issued every week, generally requiring one or two programs to be written; these assignments include preparing flow charts or pseudocode and composing memos or letters showing comprehension of the results. In-class lab assignments are given for the students to complete during a single class meeting to allow them to practice material quickly and competently. There are consequently many elements for the students to complete throughout the course that give them opportunities to demonstrate their mastery and understanding of the course concepts – quizzes, homework, and Lab assignments –while under some pressure for quick solutions. The quick turn around period trains the students to complete their work more efficiently –and start early. This tends to better prepare them for the final project, which does require them to find and evaluate its solutions in a limited timeframe.

**Planning and Development of the Integrated Project**

Similar to past engineering programming courses, seven faculty members taught the Engineering Problem Solving with Computation course and met once every second week to discuss course issues. As noted above, the faculty were motivated to develop and adopt a more authentic method of final assessment. As a first step, the entire group provided input to generate a template of topics. This was accomplished in two meetings and by e-mail and the resultant matrix is shown in Table 1. This matrix identified the elements and topics that were rated highly by each faculty member, and showed what each person felt should be included in projects that were to be designed. These calculations then determined what became required elements and topics for all classes, contributing to course consistency across instructors. Therefore, the projects all required use of both C++ and MATLAB, contained certain elements such as functions, arrays and loops, and required data or information to be exchanged between the two programs in the form of output and input files. Furthermore, all agreed that a critical element to be evaluated was the distinction between the capabilities and advantages of each type of software. For example, MATLAB is better at managing matrices and arrays and has plotting capabilities; C++ excels at computation and logic. Students therefore were required to recognize these features in order to select and properly apply the tools.

Once the matrix of topics was established, instructors developed projects in their areas of expertise, providing instructions and solutions. These drafts were distributed to the faculty team, reviewed, clarified and refined, and formatted prior to becoming available for adoption by each instructor. This review process helped prevent some of the unforeseen clarifications typically required the first time that an assignment or project is given to students. Another outcome of this process was a deeper appreciation on the part of each faculty member for the other engineering-related disciplines and their suitability for project candidates. The project write-ups were reviewed and revised by the team, verifying that the projects were reasonable, understandable, and had the required elements and applied programming to an actual engineering situation.
Integrated Project Topics and Descriptions

The standard project generally required students to show evidence of a problem formulation or plan, properly select and use both SOFTWARE tools to solve the problem, correctly generate results and output, and report their findings accordingly.

**Industrial Engineering – Intermodal Container Volume and Profit Optimization.** This is an industrial engineering application requiring students to write programs to sort, filter, and prioritize shipping order candidates. They establish delivery feasibility on three distinct criteria: fitting into transport container, percentage of vendors represented in a shipment, and profit (amount of revenue for each item and order shipped).
Students are provided with a spreadsheet of potential multiple-vendor shipping orders with various combinations of materials to be transported in intermodal containers. Their task in C++ is to import the data from the file, identify which orders will fit (by volume) into the available container and meet the vendor percentage criteria (abide by a prespecified combination of vendors), using control structures. Next, these order details are sent to an output file to be read by MATLAB. In MATLAB, matrix manipulation will be used to determine which of the candidates will yield the most profit and this information is presented in an output report. Students then make a business proposal to recommend the best combination of orders to fill.

**Mechanical Engineering – First order Model of a Thermocouple Junction.** This is a mechanical engineering example to design a thermocouple transducer that is to be used in a measurement process. Students must develop a first order model to describe the transient behavior of the thermocouple using C++. Once the model is verified for accuracy, the students will use it to vary a number of input parameters such as material, junction size and heat transfer coefficient to develop performance curves. Output from the C++ program is imported into MATLAB where it is plotted and curve-fitted. Students use the information gained from the performance curves to design the thermocouple to meet the experimental requirements.

**Electrical Engineering – Error-Correcting Code for Data Transmission.** This project is an electrical engineering simulation of a complete error-correction system via a simple linear block code. Such block codes are widely used in mobile wireless communication, satellite transmissions, optical and magnetic recording devices (CDs, DVDs, cassettes), to name a few. Each student was randomly assigned a unique linear code: the examples used in previous years included both Hamming Codes and Repetition Codes.

Students are required to generate a large number of random messages or "data" vectors. These raw data vectors are then "encoded" using the linear block code assigned to them. The resulting "code words" are then transmitted over a binary symmetric channel with varying error probabilities. Finally, these newly received "noisy" code words are decoded and the subsequent decoding errors that occurred are recorded. In this way, the students calculate the "bit error rate" (BER) of their particular code for this particular channel model. The students write their own code to do all of the above steps in both MATLAB and C++ and determine which version completed the simulation faster. Finally, they are required to plot their BER curves using MATLAB and report the results.

**Mechanical/Environmental Engineering – Flight Simulator Wind Speed Project.** In this project, the students model and analyze information on wind speeds for a flight simulator. They are to generate wind speed data randomly over time, and add storms and other disturbances to the wind speed data, using C++. Their program must generate storm data given a probability of a storm occurring, and similarly for a microburst (intense storm). They must use arrays to store this data, which is then input data for MATLAB to be plotted. Functions are written to compute average wind speed and standard deviation of the speeds. Output is the averages for various conditions, and plots of wind speeds over time. The objective is to have students generate data using random number generation and probability, analyze the data and export the data for?
MATLAB input and plotting. This project is adapted from the MATLAB textbook by Etter and Kuncicky (2002).

**Computer Engineering – Parts Tracking Program for Wafer Manufacturing.** Students are asked to design a production tracking system for computer wafers manufactured in a clean room. Results from each testing station are simulated and stored in two-dimensional matrices using C++. Results are retrieved for manufacturing management, analyzed and displayed graphically using MATLAB.

**Physics – Numerical Modeling of Planetary Motion.** Students are asked to develop a numerical model to describe planetary motion while taking into account the variation of a planet’s acceleration as it orbits the sun. The students are provided with the fundamental equations governed by Newton’s Laws of motion and are asked to develop algorithms in both C++ and MATLAB. They are then asked to run various cases to determine advantages of one software program over the other, as well as to test certain approximations used in the study. Results are presented as comet plots in MATLAB.

**Administration of Integrated Projects**

By agreement, the integrated projects were assigned during the second-to-last week of class. To overcome the issue of collaboration and cheating on this final project, instructors selected different sets of projects from the set of projects described above. Within a section, each instructor judiciously handed these out to reduce the number of friends and/or Pair Programming Partners with the same project. Instructors selected two or more projects to assign in their classes from the available project set. The limited number of projects made the evaluation procedure clearer and more consistent, which would be more difficult with one very open-ended project. Conversely, too many projects would make it difficult for the professor to master the nuances of all assignment details.

While the projects were distributed to the class, in-class time was granted for students to clarify instructions for their given projects. Over the following week, supervised lab time was also provided for project work with the proviso that the instructing faculty would only be able to assist in clarifying instructions and address problems with the computer hardware or software. This also provided an excellent opportunity for the faculty to observe first-hand some of the students’ work habits and problem-solving skills.

**Grading the Integrated Projects**

Whenever possible, we recommend generating and referring to a grade guide when evaluating the projects. Figure 1 shows an example of the grade guide used for the Industrial Engineering Intermodal Shipping Project. All criteria are drawn from project instructions and as such grading is more consistent.

In contrast to an exam format, the integrated project is clearly more open-ended and allows for a more customized evaluation to be made. More unconstrained time is available and therefore more qualitative questions can be posed. Students are required to solve a problem that has a
proper method, but which may have multiple solution paths that could be followed to accomplish the same or similar outcome. Therefore, the instructors can see the uniqueness of each student's work in their chosen approach to the problem. The level of permitted discretion is typified by an excerpt from instructions in the intermodal shipping project which reads, "The rest is up to you: make a recommendation of which batch to ship as the best business decision for [the company]. How you approach this project, format the output, and present and report your results are at your discretion."

When the time comes for the project to be graded, the code is executed to verify that it works and creates the desired results. The instructor further evaluates the planning, code, documentation, and outcome report or letter. This assessment will establish whether a student has understood the course and software concepts. It is crucial to note that the presentation criteria for results have been set out throughout the semester, as students are always required to demonstrate their comprehension in writing and by displaying their findings in tables. This expectation is not relaxed for the final project. Finally, as in Figure 1, the students are provided with opportunities to improve their grade by enhancing or adding features and capabilities to their programs.
Quantitative Outcomes

Three experienced instructors compared the grades on the integrated project with those of their past final exams and final quizzes for previous comparable courses covering C++ and MATLAB separately. The C++ and MATLAB courses, homework assignments and quizzes were administered in an analogous manner in each of the courses by these same professors. Course sections were required to keep time logs for their time invested on the project. Since these students had been keeping time logs for other assignments in the course, this task was familiar to them and the record-keeping process had been refined throughout the semester.

Grades. Since the integrated project was motivated by a conversion to semesters from quarters, no direct comparison could be made with prior combined MATLAB-C++ courses. However, both of these applications were covered in previous engineering problem-solving courses in the first-year engineering curriculum and the past grade distributions were available.

In a preliminary comparison of professors for each type of assessment, Analysis of Variance (ANOVA) revealed no significant differences in grading averages across MATLAB quizzes, no differences in C++ exam scores, nor significant differences in Integrated project grades among any of the professors at the alpha=0.05 level of significance. The premise that the instructors graded exams and quizzes comparably is thus supported. As such, the data for individual sections of the courses were collapsed across the factor of instructor and combined to compare the three assessment outcomes for grade differences.

A single-factor ANOVA was conducted to compare the course assessment outcomes to one another: MATLAB final quiz scores, C++ final exam scores, and Integrated Project grades. The ANOVA revealed that significant differences did exist among the three levels of course assessments ($F_{2, 855} = 25.95, p < .001$). Post-hoc pair-wise analyses with Bonferroni correction showed that grades on the Integrated Project were significantly higher with a mean of 89.8±11.6 than those of the MATLAB quizzes which had a mean of 85.59±14.0 ($p < .005$). Additionally, the mean grades on the Integrated Project were significantly higher than those of the C++ final exams, which had a mean of 81.18±11.9. These results are plotted in Figure 2. Note that for each of the 3 outcomes, there were comparable provisions to earn over 100% due to the potential point values on quizzes, exams, and the project.

Comparing grades on exams to grades on projects, as shown above, may be considered with some qualifications. In order to comfortably make this comparison, it is important to refer to Table 1. This table contains all potential topics covered in the courses. In order for the comparison to be valid, the topic coverage must be the same or at least similar. Exam topics from the complete list in Table 1 were reviewed for each project, to insure that the topics that were formerly on exams were now included and graded in projects. Each element of the project solutions, even with individual student variation, was an exam element and vice versa. The difference between exams and projects are in the time and increased complexity allowed on a project. Therefore, with more challenging problems, including integration of two software programs (which was not on exams), it might even be expected that students would have poorer performance, which is not the case.
The results show that when the students are given a real-world problem -adjusted to their capabilities- and are provided sufficient time to develop its solution; they perform better on average. They were able to achieve a specified goal by following a multiple-step process, within parameters, by a deadline. This is comparable to the environment in which engineers naturally thrive, being problem solvers with challenges to overcome in an “efficient timeframe”. It is an outcome we were hoping for, that students could undertake projects, solve problems, explain the results adequately and on time, more effectively and in accordance with their personal level of skill and understanding than taking an exam or quiz.

**Time Investment.** The records kept by the students on time invested in their projects included time in the lab and time spent composing, analyzing and reporting the results to the nearest quarter hour over a 10-11 day span. These values are plotted below in Figure 3. From the available data, no significant differences emerged in the time spent on each of the different versions of the projects, suggesting that assignments and their directives were comparably challenging. The average time spent on an integrated project was reportedly 9.75 hours +4.6, ranging from 3 hours to 24 hours. The median time investment was 8.5 hours.
While no significant correlation was found between the recorded time investments and earned grade on the integrated project, some relevant observations were made. The highest investments of time, 23-24 hours—which were outliers—were reported by students earning grades in the lowest category (bottom 10%), indicating that regardless of the time spent, these students could not achieve the desired accuracy or quality. In this instance, we can confidently suggest to the student that their final grade, albeit low, properly reflects their understanding of the course constructs, as they were given full opportunity to display their talents for assessment. We believe this provides more “legitimacy” and acceptance of the final assigned grades. In addition, a single student spending only 3 hours on the project earned the lowest grade. Accordingly, the trimmed statistics with the outliers eliminated, results in an adjusted time investment of 9.1 ±3.0 hours. It is suggested that when assigning the project, students are advised of these averages for their own planning purposes.

**Insight from Time and Grade records:** If the integrated project is to be an adequate assessment tool, it needs to show that students who have not mastered the material, even with ample time, will not earn an outstanding grade. This was true in many cases, as extra time did not always translate to a high score. On the other hand, doing the work quickly or with little effort does not typically result in an excellent grade; only a rare student achieves that. Some students will take extra time to review, improve, and document their code and it is important to allow for their grade to reflect this effort; exams cannot always reveal this level of diligence. A time investment of 8-9 hours indicates that there is more for students to do than can be done in two hours, but this is not an unreasonable expectation of students over the permitted time span.

Overall, the students responded well to the projects. Grades on the project ranged from below 60 to over 100 (due to extra credit offered), with the averages near 85 for non-honors students, and 94 for honors students. There did not appear to be a strong correlation between time spent on a project and grade. Highly skilled students spent less time on the project, while weak students sometimes worked long hours to still receive only a fair grade. For many cases, more hours resulted in a better grade on the project because more detail was provided in these projects. Also, in a large majority of cases, the score on the project was as good as or better than the relevant student’s final grade or course average coming into the project. The projects were generally of very good quality, including the written reports that communicated and interpreted the results.

**Faculty Reaction.** Comparisons between the two primary methods of assessment—exam/quiz and project—indicate that there is no appreciable difference in workload for faculty, yet the integrated projects are more interesting to grade. While professors can always share and combine questions for an exam, the project problems can now be more complex and interesting and not as trivial as was the case in past exam questions. Additionally, the project experience is more relevant to the students’ future professional work and therefore more interesting and motivational to them. Also, as previously stated, in the integrated project there is no guessing by the professor as to whether a student’s code is going to successfully compile and run as opposed to the final exam, in which the instructor is required to forecast the success of a written program.
Academic dishonesty in the form of copying or borrowing source code is naturally a concern; however, time constraints, workload, and the opportunity to earn points with extra quality and features make it unlikely that students help one another. In fact, due to the complexity of the problems and the ability to customize the solutions, it would be more likely that an instructor will spot unauthorized collaboration on the integrated project than on a test. Since final exam questions are designed to extract as much information as possible in a short amount of time, they are more likely to have common features and standard responses making cheating less obvious and more difficult to prove.

**Student Feedback: ‘Everybody wins’**. In response to an open-ended question at the end of the project and in remarks in their correspondence, students stated that they preferred working on real projects, and not having the extra pressure of an exam. Anecdotally, they expressed a marked lack of interest in a take-home exam as an alternative. They preferred being able to show their skills with an applied problem and being able to legitimately claim computational integration with two software applications. Since students were better able to show their “wares”, they felt that that they got more out of the course. On the whole they recognize that this is more advantageous to them, subsequently the course earns a better reputation and higher evaluations. Given that the projects are more meaningful to the students, they gain a practice-oriented education and are more attractive to hire. This, consequently, then makes the university more attractive to future students.

A sampling of student quotes summarizes the positive feedback and general attitude of the majority of the students:

"Over the course of this class, I have learned many things about MATLAB and C++ that I was not familiar with before. The programs I have written for this semester’s final Integrated Assignment clearly demonstrate my ability as a programmer, as I have put my best effort into the code. I have enjoyed this class and feel that I have learned a lot from this experience."

"This assignment was exciting and challenging...Since careful planning and time have been invested into these two programs, they can be modified to complete other tasks without starting from scratch [and are] designed to be as versatile as possible while attaining the desired goals."

"I was very surprised by how easily I was able to combine the two programs used. It was interesting to find myself using two different programs to complete the same assignment. Overall, the project was a better idea than a written final exam in my opinion"

"The project proved to be very useful because it tied the whole semester together in one project. The project itself was interesting enough that it overtook some of the tedious parts. This project also refreshed my mind of the C++ I had previously learned I have taken quite a bit away from this class, and even plan on using some of the skills in my summer job."

"There were parts more difficult than others, but on the whole this project was definitely at the skill level of a programmer like me. This class has given me an excellent knowledge of both C++ and MATLAB."
Discussion

Include Authentic Features of Corporate Engineering. In engineering practice, some computational work is accomplished on paper, but not in the same fashion as is required in exams. The integrated project requires planning to be done (hand-written planning and staging) which is real-world “paperwork”. In the integrated project, students are evaluated on this work, which may take the form of flowcharts, algorithms, or pseudocode. Exam time constraints do not allow the opportunity for much quality planning, which is unfortunate as this is how we recommend programs to be developed throughout the term. The integrated project reinforces this requisite for quality planning, in keeping with the course and standard practices in the profession. In all course assignments, the students must generate memos, reports, summaries or letters, cover sheets, screen captures, and other results, which are not possible to produce in an exam. The students must also be able to interpret and report results, which is a better test for real-world competencies Therefore, an exam is a suboptimal means of assessing both the planning and presentation aspects of the students’ work. It is in its applicability and relevance to the real world that the integrated project’s true value is revealed.

Fairness in Grading and Student Confidence. All students want to feel that they have been evaluated fairly. The opportunity for them to provide more detailed answers provides for better understanding of their knowledge, so that one bad [exam] day is not the “end of it” for them. Students are given the time and opportunity to demonstrate sound, thoughtful, and meaningful considerations about their work and can actually choose how they want to represent themselves, as they would in the workplace. As instructors, we have a clearer idea of what a student will produce having some time to consider, decipher, and comprehend how they will approach and present their work. It will be unlikely that in their future they will be faced with small, exam-like time limits to produce results. They will be likely to have time to consider a reasoned and planned approach, perform analysis and interpretation with longer deadlines. It is this “reasoned” timeframe upon which it is fairest to evaluate students and which equally fosters confidence in their final result. The resultant realism in the problem is motivating for the students, as there is a purpose to finding a solution and there is an element of respect shown for their knowledge in the type of problem that is posed to them. Furthermore, all of the instructors administer frequent written quizzes and can attest that the final exam does not provide a new level of assessment, only a confirmation of quiz performance as was seen over the duration of the semester.

More Opportunity for Extra Credit to be Earned. As there are some open-ended aspects to each project, it allows the students to do some extra work to earn additional credit. They can accomplish things like generate plots and graphs, optimize a system, or design an interface for others to use. This additional credit need not be a skill-based activity; a student can do additional presentation improvement work in the time allowed to show dedication and effort, possibly learning a new skill. This is meaningful to include in our assessment, as employers would want to know whether a student has these attributes. This extra work shows they are inclined to do it, are able to do it, and are willing to do it. This level of knowledge, inclination, and effort cannot be demonstrated as readily on time-constrained examinations, although there had been provisions to do so in past programming-based exams.
Cross-Software Communication: As our course covers both C++ and MATLAB; it was desired that the students demonstrate the ability to use both programs, as they are best suited, and to integrate the programs to accomplish the assigned problem objectives. It is important to have a means of evaluation that examines the student’s ability to solve across problems, to make two systems communicate and to show insight choosing the correct tool. These are valuable skills they will be using in the real-world, and not necessarily only applicable with MATLAB and C++. The grade received should reflect whether the student has the skills to adapt to this situation. Throughout the freshman year, the instructors emphasize that various software tools have been designed with different capabilities and functionality. The students are taught to select the right tool for a particular job. We reinforce the concept, not the tool; attempting to instill in the students the reason that one tool is preferred over another. We have stated to them that “you can open a can with a screwdriver, but it might not be the best tool for the job”. Having taken the time to teach students these concepts, it is only fair to assess their acquired knowledge of them. The integrated project accomplishes the objective of having the students demonstrate their skills with both programming tools and in integrating the two programs, after evaluating the best use of each. The exam method does not really address these issues at all. The time spent and quality achieved was successful for this team to continue development and use of integrated projects.

Conclusion

As noted, incorporating advances in computer programming into engineering curricula has brought with it new priorities when assessing students’ work. The concept of administering a common final written exam for all students was developed when the majority of professional work was largely assembled and completed on paper and course instruction relied mainly on in-class instruction and board work for demonstrations. A written final exam was therefore a logical extension of this model. Now just as some hand calculations and written work are rapidly moving towards the “obsolete basket”, so too are its accessories, including the written examination. Our team in the College of Engineering at Northeastern University has revisited all aspects of our assessment methods and adapted more fitting evaluation techniques to address the priorities of practical application and concerns of emerging technology. “Engineering Problem Solving with Computation” was one such course that became part of our program’s elective review process and is the subject of this paper.

Changing With the Times. With the need for adaptation in mind, we elected to introduce the concept of a final integrated project to replace the final examination in our problem solving with computation course. There were many elements of the computation requirements of the course that were no longer being met by the written examination process. It did not allow the students to fully develop complex ideas, it did not provide the students with the opportunity to display their unique skills and understanding of the course requirements, nor were we able to properly assess the successful nature of their proposed solutions, as they could not be compiled or executed. In considering the option of an integrated project, we found that most if not all of the past concerns in adopting the final examination method could be almost entirely and at least satisfactorily met while simultaneously addressing our new concerns. More importantly, the final project model, which has been set out in this paper provides added benefits for the engineering students and engineering program alike.
Integration and Operation. Although our computation course uses two programming software tools to teach the course precepts; our observations of the Integrated Project in practice causes us to believe that this can be a valuable teaching, learning, and assessment instrument for any course that adopts either or both of these software tools. Concerns about increased opportunity for academic dishonesty have not eventuated but we have witnessed better quality work being produced by the students and this has made us better able to differentiate and assess their various skill levels at the conclusion of the course. Furthermore, students must have a functioning program; it is not just effort and knowledge, it must run. This is the ultimate evaluation of the students’ ability - does it actually work? As noted on written exams, it is difficult for the instructor to judge the ability of a given program to compile and execute; therefore evaluation is difficult and unrealistic for programming. While assessments are also made for all other components of the assignment, the integrated project is easily evaluated, since the bottom line is that a program that operates to solve the problem at hand is a workable solution.

Applied Evaluation Methods. The fact that Northeastern and the College of Engineering emphasize problem-solving approaches that will translate into practice at co-op jobs and in future projects is reflected in and complemented by the course and the integrated project. Students are likely to frequently encounter projects with deadlines in their future work whether programming or other engineering analysis. They must master their own time management, seek help when needed, read and research unfamiliar techniques or topics and overcome obstacles in order to complete their projects. The integrated project’s demands are compatible with adult, career, and vocational education. This type of practice is a form of apprenticeship, which is “is a time-honored form of authentic learning: i.e., skills taught in context (Huba & Freed, 2000).

The integrated project emulates problems and work requests that may be posed to the students in their professional lives and as such more closely resembles the students’ future roles in the workplace, rendering this course a “marketable” attribute in the engineering program. High-performance workplaces "demand critical thinking, self-directed learning, and individual responsibility for career development” (Borthwick 1995) The integrated project relates to current trends in educational practices that are making similar strides to ensure engineering education becomes increasingly more relevant and related to the real world and its practices and poses the question as to whether this is the way of future for engineering education.

Heading in the Right Direction. Initially we have gone from closed book to open book evaluation methods. Now accelerated advances in technology oblige educators to also adapt grading methods in a like fashion. We should continually be reviewing all aspects of our assessment and work practices. The project approach would certainly work in courses covering C++ or MATLAB languages separately, but the combined elements of this course make the integrated project a particularly meaningful and authentic evaluation method. Like using Pair Programming -which is also a reflection of real-world practices- so assessment practices should be aligned to the real world as well.
References:


Author Biographies

Beverly Jaeger, Richard Whalen, and Susan Freeman are members of Northeastern University’s Gateway Team, a selected group of faculty expressly devoted to the first-year Engineering Program at Northeastern University. The focus of this team is on providing a consistent, comprehensive, and constructive educational experience in engineering that endorses the student-centered and professionally-oriented mission of Northeastern University.