

Management Challenges in a Large Introductory Computer Science Course

A.T. Chamillard
Department of Computer Science
U.S. Air Force Academy, CO 80840
703-428-0528
achamillard@hq.dcma.mil

Laurence D. Merkle
Department of Computer Science
U.S. Air Force Academy, CO 80840
719-333-3131
Larry.Merkle@usafa.af.mil

Abstract

Many colleges and universities throughout the world offer introductory computer science courses with significant student enrollment. There are certainly plenty of challenges for the administrators of those courses; ensuring equitable grading across different instructors and offering times for the course, accomplishing the massive material preparation and other logistical tasks required for the course, grading the large number of assessments associated with the large enrollment, and managing the numerous instructors in the course are a few examples. This paper discusses the processes we have implemented to address these and other management challenges at the U.S. Air Force Academy.

Keywords

Large courses, introductory courses, course management, introductory computer science.

1 Introduction

All students at the U.S. Air Force Academy (“USAFA”) take an introductory course in computer science in either their freshman or sophomore year. One of the key areas of study is problem solving with computers, assuming no prior computer knowledge. We start by developing the students’ general problem solving skills, then help students extend these skills to solve problems using computers and the Ada programming language.

Typical enrollment in the course ranges from 400 to 700 students per semester. Key factors affecting this are the size of the entering class and the faculty manning level. Sections are limited to 23 students to keep the student/teacher ratio low and because of limitations on laboratory facilities, which leads to 24 to 33 sections each semester, and typically 17 to 20 instructors.

There are clearly a number of challenges facing the administrators of any large course [3, 4]. One of the most important challenges is ensuring that students are graded equitably across the numerous sections of the course (see [2] for one example). To address this, we have implemented a number of grading policies and processes.

There are other management considerations as well. Administering the development and printing of the course materials, grading the large numbers of assessments without the benefit of teaching assistants, and managing the numerous instructors for the course all offer challenges.

The next section describes the processes we have implemented to try to ensure equitable grading across sections, and Section 3 presents some informal analysis of grading data. Section 4 provides details about our approach for addressing other challenges associated with managing a large course. The final section presents our conclusions.

2 Grading Processes

While it can be argued that students should accept their instructor, along with their grading tendencies, we believe that students who do the same quality of work should receive the same grade. This is particularly important at USAFA, because overall GPAs affect our students’ class standings, which in turn have a direct effect on their job opportunities upon graduation. Thus, we try to ensure that students are graded equitably no matter which instructor they have. Perfection in this regard is not possible, but this section describes our efforts toward that goal.

One area in which we can place the students “on a level playing field” is in their preparation for standard testing activities: two tests and the final exam. At the beginning of the semester, we provide students with a list of objectives for each of the 42 lessons in the course, categorized according to Bloom’s Taxonomy of Educational Objectives [1]. This helps them identify the important ideas from each lesson and indicates the level at which we expect them to master particular objectives. Some example objectives are provided in Figure 1.

When we create the testing instruments in the course, we develop the questions directly from the lesson objectives. In fact, for each question on the instrument we indicate the objective from which that question was generated. Many

This paper is authored by an employee(s) of the United States Government and is in the public domain.
SIGCSE '02, February 27- March 3, 2002, Covington, Kentucky, USA.
ACM 1-58113-473-8/02/0002.

Knowledge:

1. Identify the primary Ada constructs that are conditional in nature.
2. Look up the syntax for basic loops and the EXIT WHEN statement.

Comprehension:

1. Analyze and predict the results of Ada code containing basic loops.
2. Understand when a variable must be initialized prior to a loop.
3. Understand that the Boolean expression controlling a loop, combined with modification of variables in the loop, must eventually lead to an exit from the loop.

Application:

1. Use a basic loop and the EXIT WHEN statement to implement iteration in programs.

Figure 1. Example Lesson Objectives

instructors suggest their students use the lesson objectives as a study guide when preparing for the tests.

While the objectives provide students with information about the focus of each lesson, they do not necessarily help instructors observe that focus during each lecture. To try to provide this focus, and to communicate possible presentation techniques to the instructors, we develop an Instructor Lesson Plan (ILP) for each lesson in the course. To develop the ILPs for the course, each instructor generates ILPs in a standardized format for two or three lessons; the ILPs are then consolidated to provide ILPs for the entire course. While some instructors choose to develop their own lesson plans, most instructors start with the standard course ILPs. We also note that the ILPs do not direct the instructors how to present the material, they merely provide a reminder of the lesson objectives and a suggested approach for covering the material. In addition, instructors are not compelled to cover the entire contents of the ILP if they choose not to; many instructors are comfortable reminding the students that they are responsible for all the objectives for the lesson, whether or not the lecture actually covers all those objectives.

Our next step in our pursuit of equity on the testing instruments is to provide group review of these instruments before they are administered and to provide group grading of the instruments afterward. For the group review, we assign a single instructor to develop the instrument and assign two reviewers to complete a preliminary review. The instrument contains both the questions and the preliminary grading criteria for each question. After incorporating the reviewer comments, the instructor distributes the instrument

to all instructors in the course for review. Several days later, all the course instructors convene to discuss each question in the instrument. After incorporating comments from this meeting, the instructor prints the instrument and it is administered to the students. This review process provides us with an excellent opportunity to identify problems in the instrument before it is administered.

After the instrument is administered, we convene the course instructors to grade the instrument for the entire course (which typically consumes the better part of a day). To ensure consistent grading of each question, we assign two or three instructors to each question. Those instructors finalize the grading criteria for that question, then grade that question for all the students in the course. This ensures that each question is graded consistently independent of section and instructor. Additionally, if students take issue with the grading of a particular question, they are directed to the instructors assigned to that question, ensuring that the grading consistency persists through the "post test return point skirmishes." We also attempt to include as many true/false and multiple choice questions as possible on each instrument; these questions can be graded automatically, ensuring consistent grading and freeing instructors to grade the short answer and programming questions.

Despite our efforts to make the testing instruments as equitable as possible, we note that our approach is still not perfect. For example, in both the review and grading activities we have had instructors object to a question or the grading criteria based on the "I didn't teach it that way" argument. This poses an interesting dilemma. On one hand, the instructor could revisit the topic in class to ensure the students have the necessary information before taking the test. On the other hand, requiring this could both damage the credibility of that instructor and identify some of the test contents for the students. In general, we either re-word these questions or remove them from the testing instrument altogether.

We also occasionally run into problems with the grading criteria for particular questions based on different grading philosophies. The vast majority of the time, the instructors assigned to a particular question are able to come to agreement on the grading criteria. We have (admittedly rarely) had grading issues that needed to be resolved by the Course Director for the course, however; the most severe case involved an argument from one instructor that we should give partial credit for an incorrect answer to a multiple choice question. We note that these differing levels of grading leniency are one of the main reasons we have implemented these processes in the first place!

There is also a programming component in this course, consisting of programming labs, lab practica, and a group case study. We would like to ensure these assessments, which constitute approximately 40% of the student grade, are also graded equitably. Conversely, logistical difficulties preclude our using "group grading" for these assessments as

| | |
|---|------------|
| <ul style="list-style-type: none"> ● Works (Executable requirements met/No Compile = 0, Validates Input (2), Computes Screen Height (1), Gets Mouse Position (1), Draws & Erases Circle (1), Uses Proper Color (2)) | _____ (7) |
| <ul style="list-style-type: none"> ● Partial Credit for Solution (Subproblems attempted properly (double scores above)) | _____ (14) |
| <ul style="list-style-type: none"> ● Testing (Boundary Values (3), "Other" Boolean Expression Test (1)) | _____ (4) |
| <ul style="list-style-type: none"> ● Programming Standards (Comment block with description, Reformat used, line comments, variable names, mailed properly, printout included) | _____ (5) |
| Subtotal | _____ (30) |

Figure 2. Example Grade Cuts

we do for the testing instruments. In addition, instructors focus on different aspects of programming, and to reinforce that focus each instructor should grade their own students to ensure points are given (and taken away) appropriately.

To help ensure equitable grading of the programming assignments while still having each instructor grade the assignments from their own sections, we provide grading cut sheets for each assignment. These cut sheets are distributed to the students with the assignment, so they know how they will be graded on that assignment. Instructors use the cut sheets as guidelines during their grading of a particular assignment. An example of the grade cuts on a cut sheet is provided in Figure 2.

Our philosophy for these cut sheets has evolved over time. Specifically, we have struggled with the appropriate level of granularity. At times, we have documented cuts for each point on the assignment. This certainly provides equitable grading on the assignments, but does not allow the instructor the flexibility to emphasize their focus areas. On the other hand, simply omitting the cut sheets and having instructors grade completely at their own discretion has resulted in vast differences in grades on these assignments, even when the students completed approximately equivalent levels of work. While we will continue to evolve this approach, we believe that the current cut sheets are at approximately the correct level of granularity.

3 Grading Data Analysis

The testing instruments and programming assignments described above comprise 965 points out of the 1,000 in the course; the only other points are for a learning styles survey (graded as completed or not), a web page (essentially graded as completed or not), and bonus points (beyond the 1,000) associated with completion of extra credit activities.

Recall that our purpose in implementing the above processes is to ensure equitable grading. To determine

whether or not there is a reasonable return on our considerable investment in these processes, we would like to identify if, without these processes, we would potentially observe inequitable grading (either overly lenient or overly harsh) from specific instructors.

To consider this question, we examine the grading data from the Spring 2000 offering. In this semester, there were 19 instructors teaching 30 sections of the course. We segregate the grading data into three parts: "instructor percentage," "group percentage," and "auto percentage." The instructor percentage represents the average percentage that the students for that instructor received for work graded by the instructor (340 points); note that we do not include the 75 points associated with the group case study, since most students get almost full credit on this assignment, which would only serve to raise all the instructor percentages. The group percentage represents the average percentage that the students for that instructor received for work graded by the group of course instructors (323 points), and the auto percentage represents the average percentage that the students for that instructor received for work graded automatically (227 points). The comparison table of these percentages is provided in Figure 3.

First, we note that course-wide the instructor percentage is 14.6% higher than the group percentage and 11.0% higher than the automatically graded percentage; we use these relationships as our baseline of comparison when we consider specific instructors. By using this comparison technique rather than simply comparing the instructor percentages, we consider differences in instructor grading while removing the effect of strengths and weaknesses of specific sections of the course. For our analysis, we only consider relationships that differ from this baseline by more than 5% as worthy of discussion. We first consider instructors who seem to grade more leniently than the

| Instructor | Instructor % | Group % | Auto % | Instructor | Instructor % | Group % | Auto % |
|------------|--------------|---------|--------|-------------|--------------|---------|--------|
| A | 84.8 | 68.8 | 73.7 | K | 83.9 | 63.1 | 70.3 |
| B | 89.1 | 73.0 | 72.8 | L | 86.8 | 76.5 | 80.4 |
| C | 86.4 | 73.8 | 79.1 | M | 92.1 | 74.8 | 76.6 |
| D | 93.3 | 71.3 | 72.9 | N | 73.8 | 67.2 | 78.9 |
| E | 85.8 | 67.4 | 76.4 | O | 92.4 | 77.8 | 79.3 |
| F | 84.7 | 71.5 | 73.7 | P | 82.6 | 69.9 | 73.7 |
| G | 79.7 | 68.0 | 77.3 | Q | 87.4 | 70.4 | 72.1 |
| H | 87.5 | 79.1 | 77.6 | R | 79.6 | 65.4 | 73.5 |
| I | 86.2 | 74.0 | 75.1 | S | 90.3 | 77.3 | 76.7 |
| J | 89.6 | 74.2 | 77.4 | | | | |
| | | | | Course-Wide | 86.6 | 72.0 | 75.6 |

Figure 3. Course Grading Percentage

course instructors as a whole, then consider instructors who seem to grade more harshly.

Instructor D has the most significant differences from the baseline; the instructor percentage is 22.0% higher than the group percentage and 20.4% higher than the auto percentage. Instructor K also graded more leniently than the course instructors as a whole, with an instructor percentage 20.8% higher than the group percentage and 13.6% higher than the auto percentage. Finally, Instructor B had an instructor percentage slightly higher than the baseline compared to the group percentage (16.1%), but significantly higher than the baseline compared to the auto percentage (16.3%). It seems clear that these three instructors graded more leniently than the group of all course instructors, especially Instructors D and K.

For instructors that appear to grade more harshly than the norm, Instructor N has the most significant differences. For this instructor, the instructor percentage is only 6.6% higher than the group percentage, and is in fact 5.2% LOWER than the auto percentage. Instructor H also graded more harshly than the course instructors as a whole, with an instructor percentage 8.4% higher than the group percentage and 9.9% higher than the auto percentage. Finally, Instructor G has an instructor percentage slightly lower than the baseline compared to the group percentage (11.7%), but significantly lower than the baseline compared to the auto percentage (2.4%).

We have identified a number of instructors who grade with more leniency or harshness than the group of course instructors as a whole. This implies that our processes are in fact useful, but only if some of the percentages have a large impact on the course grades for the students of that instructor. We therefore examine the correlations between the average course GPA for the students for each instructor and the three percentages presented above. We find that the average course GPA is very highly correlated, with $p < 0.05$,

with both the instructor percentage (0.887 correlation) and the group percentage (0.892 correlation). Because these correlations are so high, removing the group grading activities and having the instructors grade those instruments instead would further heighten the impact of the grading philosophy of each instructor. This, in turn, would mean that the course grades for each student would be more susceptible to the grading differences exhibited by the course instructors.

It seems clear that our processes are in fact needed to try to provide equitable grading to the students. Of the 19 instructors for the course, 6 instructors appear to grade with either noteworthy leniency or noteworthy harshness. The policies we have implemented help “dampen” those effects on the ultimate grades of the students, thereby making their grade less dependent on their instructor and section for the course.

4 Other Management Challenges

While equitable grading is one of our key areas of interest in the course, there are numerous other management issues that merit further discussion. In this section, we briefly describe the management structure of the course, then discuss several management challenges in such a large course.

The Course Director (CD) of the course has the ultimate responsibility for managing and administering the course. We have also assigned an Assistant Course Director in the past, but the use of this assistant has been at the discretion of the CD. The instructors for the course are essentially “matrixed” to the CD; they work for the CD when accomplishing tasks for the course, but do not have the CD as their supervisor.

One of the challenges faced by the CD is ensuring that the course materials are developed and distributed to the students in a timely manner. Because the course is continually evolving, simply re-using the course materials

from the previous semester is not a reasonable approach, though of course most CDs use those materials as a starting point. The work associated with developing all the materials for the course, which include lesson objectives, supplemental readings, and so on, is far too much to be accomplished by the CD alone. It is therefore necessary to distribute some of this workload to the course instructors. This helps ensure the material is developed, but it is difficult to get a consistent style within the materials when numerous people are developing them. In addition, ensuring that all instructors complete their assigned materials in a timely manner can be a difficult management challenge, especially in our matrixed environment. Finally, once the materials are developed, the CD needs to allow sufficient lead time for printing the materials - since all other courses at USAFA are trying to have their materials printed at the same time, course materials need to be provided to the printer a minimum of three weeks in advance.

Another challenge faced by the CD involves the grading load that the course imposes on the course instructors. We do not have any Teaching Assistants at USAFA, so all the course grading is completed by the instructors. The CD is therefore faced with the difficulty of including enough assessments in the course to provide students with reasonable feedback and evaluation without swamping the course instructors with an impossible grading burden. In addition, the Dean imposes a limit on the number of assessments allowed in each course; his incentive for this limit is to ensure that instructors strive to motivate their students rather than having the students motivated by "fear of daily assessments." The CD must also consider these policies when selecting the assessments for the course.

The grading load in the course is reduced by the use of the group and automatic grading techniques discussed above. While each instructor participates in the group grading for a total of approximately three working days over the course of the semester, they then only have to grade 6 programming labs, two lab practica, and a group case study. They receive some assistance in this grading as well, because we have developed some automated tools to help with the grading of both the practica and other programs the students generate during the course. Essentially, these tools run each program with a given set of inputs, generating textual outputs in a separate file which can then be checked for correct program performance.

The absence of Teaching Assistants is evident in another area as well - where do students go when they need help with topics in the course or their programming assignments? Although the laboratory assignments are collaborative, many students prefer discussing their programming problems with their instructor. Because instructors teach up to 69 students in the course, this can also result in a large time commitment for the instructor.

Finally, managing the large number of instructors for the course presents its own challenges. Trying to accommodate

the teaching preferences and scheduling conflicts for all the course instructors can be difficult. Communicating the course goals and policies to the course instructors can also be difficult, though the CD typically holds a weekly course meeting to discuss policies, presentation ideas for upcoming course material, and any issues that have arisen in the course. The CD is also rarely faced with situations in which an instructor has violated a course policy (for example, that late turn-ins are not accepted without prior coordination); resolving those situations while maintaining the instructor's credibility can be a challenge as well.

5 Conclusions

There are many challenges associated with developing and administering a large introductory course in computer science. Ensuring equitable grading across instructors and sections in the course is a primary consideration for us at USAFA, so we have implemented processes designed to help us meet that goal. There are other management challenges as well, of course, and we have described our approach to some of those challenges above.

Although there are some interesting and difficult challenges associated with managing a large group of course instructors, we actually enjoy a very high level of cooperation and teamwork among the faculty in our department. Many of the challenges occur because we set very high goals for the course and its administration; while we sometimes face some difficulty meeting those goals, we are able to pursue them in the first place because of the collegial atmosphere in our department. Of course, the computer science department at USAFA is not unique in this regard, so the processes and goals described above could also be adopted for use in other departments at colleges and universities throughout the world.

References

- [1] Bloom, Benjamin S. (ed), *Taxonomy of Educational Objectives: Handbook I Cognitive Domain*, David McKay Co., 1956.
- [2] Chamillard, A.T. and Joiner, Jay K. Using Lab Practica to Evaluate Programming Ability. In *Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education*, Charlotte, North Carolina, February 2001.
- [3] Kay, David G. Large Introductory Computer Science Classes: Strategies for Effective Course Management. In *Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education*, Atlanta, Georgia, February 1998.
- [4] Kay, David G., Carrasquel, Jacobo, Clancy, Michael J., Roberts, Eric, and Zachary, Joseph L. (panel presentation) Managing Large Introductory Courses. In *Proceedings of the Twenty-eighth SIGCSE Technical Symposium on Computer Science Education*, San Jose, California, February 1997.