Individualized exercises are a promising kind of educational content in modern e-learning. The focus of this paper is the QuizPACK system, which is able to generate parameterized exercises for the C language and automatically evaluate the correctness of student answers. We introduce QuizPACK and present the results of its comprehensive classroom evaluation when used for four consecutive semesters. Our studies demonstrate that when QuizPACK is used for out-of-class self-assessment, it is an exceptional learning tool. The students’ work with QuizPACK significantly improved their knowledge of semantics and positively affected higher-level knowledge and skills. The students themselves praised the system highly as a helpful learning tool. We also demonstrated that the use of the system in self-assessment mode can be significantly increased by basing later classroom paper-and-pencil quizzes on QuizPACK questions, motivating students to practice them more.

Categories and Subject Descriptors: K.3.1 [Computers and Education]: Computer Uses in Education - Distance Learning; K.3.2 [Computers and Education]: Computer and Information Science Education - Self-assessment
General Terms: Human Factors, Performance, Languages
Additional Key Words and Phrases: E-Learning, Individualized Exercises, Parameterized Questions, Assessment, Code Execution, Introductory Programming, Classroom Study

1. INTRODUCTION
Web-based quizzes became the primary tool for assessment and self-assessment of student knowledge in the context of Web-based education [Brusilovsky & Miller 1999; Brusilovsky & Miller 2001]. All major Web learning management systems (LMS) support authoring and delivery of on-line quizzes made from static questions. LMS vendors spent a significant effort extending the range of question types, making question-authoring easier for the teacher, and providing support for managing quizzes and pools of questions. Yet, even with these modern, powerful authoring tools, the authoring of Web-based questions and quizzes remains a difficult and time-consuming activity. As a result,
the number of questions in a typical Web-based course remains relatively small. This creates problems with using Web-based questions in both assessment and self-assessment modes:

- Within an assessment, the lack of questions results in cycling through the same or nearly the same set of questions for each student in a class (and often for different classes and/or different semesters). This arrangement invites cheating, even in a controlled classroom assessment context. For take-home and Web-based assessment cheating becomes a major problem, often preventing teachers from relying on results from this kind of assessment.

- In self-assessment, where cheating is not a problem, the number of available questions is typically insufficient for the students to assess the level of their knowledge and in order to guide further study.

A known remedy for this approach is the use of parameterized questions and exercises. A parameterized question is essentially a template for the question, created by an author. At presentation time, the stated template is instantiated with randomly generated parameters. A single question’s template is able to produce a large or even unlimited number of different questions. Thus, during assessment, a reasonably small number of question templates can be used to produce individualized assessments for large classes. Moreover, the same templates can be used in different versions of the same course, different semesters, and even different courses. In self-assessment, the same question can be used again and again with different parameters, allowing every student to achieve understanding and mastery. Cheating-proof exercise templates become reusable, non-devaluing assets, which can be accumulated over courses and years in reusable assessment libraries/pools.

Individualized exercises were explored in the field of educational technology and came back into focus recently as one of the areas of promising research in the field of Web-enhanced education. A number of pioneer systems such as CAPA [Kashy et al. 1997], WebAssign [Titus et al. 1998], EEAP282 [Merat & Chung 1997], or Mallard [Graham et al. 1997] explored the use of individualized exercises in different contexts. CAPA, the most influential of these systems, was evaluated in a number of careful studies [Kashy et al. 2001; Kashy et al. 1997], providing clear evidence that individualized exercises can significantly reduce cheating while improving student understanding and exam performance.

The main challenge of parameterized questions and exercises as an educational technology is the evaluation of student answers. A traditional static question can be
evaluated simply by comparing the student answer with a set of answers provided by the author. However, a parameterized question requires a run-time "domain expert" that can find the correct answer for each instantiation of the question. As a result, almost all known applications of parameterized questions and exercises were developed for physics and other math-related subjects where a correct answer to a parameterized question can be calculated by a formula that includes one or more question parameters.

This paper reports our attempts to implement parameterized questions for assessment and self-assessment of student programming knowledge. Assessing programming knowledge is quite different from knowledge assessment in math-related subjects and so the use of individualized questions has not yet been systematically explored in the field of programming. We describe a specific approach for the automatic generation and evaluation of parameterized questions for programming courses and the system QuizPACK that implements this approach for the programming languages C and C++. We also report on our experience with the system and present the results of formal classroom evaluations of QuizPACK during four consecutive semesters with more than 150 students. In brief, our data shows that parameterized questions used in a self-assessment mode are an exceptionally powerful learning tool that can significantly increase the student’s knowledge of programming. This tool is also highly appreciated by students. Simultaneously, we investigated how the value of this technology can be increased by creating a context that motivates students to use it.

2. QUIZPACK: THE APPROACH AND THE SYSTEM

While analyzing several large pools of questions created for classroom courses, based on languages like Lisp, C, and Java, we found that a large portion of questions in these pools are code-execution questions. In a code-execution question, the user is provided with some fragment of a program and is asked to predict the value of a particular variable or a string if it were printed at the end of execution of this fragment. These kinds of questions are very important in that they enable the teacher to check the student's understanding of the semantics of programming language structures. Learning the semantics of a programming language is an important part of programming knowledge in itself and is also a prerequisite to higher-level programming skills.

In the pools that we analyzed, code-execution questions were authored in the standard, multiple-choice style by specifying the text of the question and a set of possible answers. The student answers are evaluated by comparing them to the correct answers provided by the author. However, it is known that code-execution questions allow a
different approach to evaluating the correctness of student’s answers. Using a language interpreter or compiler, the system can run the given fragment of code and calculate the correct answer without the need of a teacher or author to provide it. Then the calculated answer can be compared with student’s answer. In the past, we used this approach in several earlier systems such as ITEM/IP [Brusilovsky 1992], ILEARN [Brusilovsky 1994], and ELM-ART [Weber & Brusilovsky 2001], in order to save some sizeable amount of question-development time and to avoid authoring errors.

When developing our system QuizPACK, (Quizzes for Parameterized Assessment of C Knowledge), we attempted to combine our earlier work on automatic evaluation of code-execution questions with the idea of parameterizing the questions. Our goal was to create a system that would allow teachers to create parameterized code-execution questions and quizzes in a streamlined way, rather than creating the same questions and quizzes over and over again in a static form, with the current authoring tools. QuizPACK functions in a very straightforward way. A teacher provides the core content of a question: a parameterized fragment of code to be executed and an expression (usually a variable) within that code. The student answers with the ending value of that expression. The system does the rest: it randomly generates the question parameter, creates a presentation of the parameterized question in a web-based quiz, receives the student's input, runs the parameterized code in order to generate the answer, compares this result to the student’s answer, gives a score to the student, and records the results.

The key element here is the ability of the system to execute the parameterized code fragment in real-time, while the student is taking the quiz. There are at least two ways to do this. One way is to use a specially designed interpreter or code evaluator that can execute the code of any exercise at runtime. Another way is to use a standard language compiler. Using a simple code transformation, the exercise code can be automatically converted into a function that takes the parameter value and returns the value to be checked. This function can be compiled into a CGI program that will compare the student’s answer to this value, at runtime. In this way, each exercise receives a dedicated checker, automatically constructed and compiled when authored.

The first method can provide additional educational value by showing the visual execution of code or by enhancing the exercise with explanations. This method was used in ITEM/IP [Brusilovsky 1992], ELM-ART [Weber & Brusilovsky 2001], and a number of systems developed by Amruth Kumar [Dancik & Kumar 2003; Fernandes & Kumar 2002; Shah & Kumar 2002] to explain to students how the correct answer to the exercise is produced. The negative side here is the need to develop such an interpreter. While it is
relatively easy for systems dealing with small languages (such as ITEM/IP for Turingal) or with language subsets, it becomes a challenge for a system that intends to support an unrestricted professional language such as C or C++. For this reason as well as for better efficiency (which is critical for a Web-based system that is simultaneously accessed by multiple users at the same time), QuizPACK uses the second method.

The current version of QuizPACK can support a complete programming-related course based on the C or C++ programming language. It supports simple questions based on a few lines of code, as well as very advanced questions that include multiple functions and header files. Figure 1 demonstrates the student interface of QuizPACK. A parameterized question is presented to the student in a browser as though it were a normal static fill-in question (Figure 1, left). One or more constants that the student can see in the body of the question are, actually, instantiated parameters. They are randomly generated for different students taking the quiz as well as for the same student attempting the question several times. The student fills in the answer and hits the "submit" button. In response, the system generates an evaluation screen for the student (Figure 1, right). This screen lets the student re-think the question and the answer. In particular, the student may want to attempt the same question again by using "repeat last question" link. The same question can be attempted several times with different presentations, until it is fully understood and, as our study shows, many students actively use this opportunity.

In the context of this journal, which focuses on resources to enhance Information and Computer Science Education, it is important to stress that QuizPACK is not only an innovative tool, but also a reliable practical system. It is available for both online use and free download at http://www2.sis.pitt.edu/~taler/QuizPACK.html and we welcome the readers to try it. Both online and downloadable versions are provided with a large number of questions, covering a complete introductory programming with C course. The downloadable version also includes a toolset for authoring new questions. With this toolset, it takes just a few minutes to turn a code fragment into a QuizPACK question. Further details about QuizPACK authoring support and implementation are outside of the scope of this paper and can be found in [Pathak & Brusilovsky 2002].
3. THE EVALUATION OF QUIZPACK

3.1 The Summary of QuizPACK Use at the University of Pittsburgh

QuizPACK has been used at the University of Pittsburgh since 2001, to support the teaching of programming courses to undergraduate Information Science students. During these years we tried this system in several contexts and run a number of classroom studies. The first version of the system was developed in 2001 and was used for two semesters in a Data Structure course based on the C language. The system and its first limited evaluation were reported in [Pathak & Brusilovsky 2002]. In 2002, the School of Information Sciences introduced a new undergraduate Introduction to Programming class that became the main framework for QuizPACK use and evaluation.

QuizPACK was originally developed for both assessment and self-assessment of programming knowledge. However, we soon discovered that the nature of the system prohibits its application as a reliable out-of-class automatic assessment. In an uncontrolled environment, resourceful students can simply compile the code of the question or run it in a debugger to find the answer. This way of solving the problem has certain educational value, but it prevents the use of the system as a reliable take-home or Web-based assessment. Obviously, QuizPACK could be used for an assessment in a controlled environment such as a classroom quiz or an exam, however, it requires a large computer laboratory (40-50 computers for our typical classes), which our school does not have.

Since QuizPACK was designed to work even on handheld computers equipped with a Web browser, it can be also used for assessment in a wireless classroom where all students are equipped with wireless laptops or handheld computers. We actually tried this way, distributing wireless HP Jornada™ handhelds to students before each class, but gave it up after two attempts, due to technical problems. The follow-up questionnaire showed that almost all students preferred traditional paper-based classroom quizzes to Jornada-based quizzes. It is interesting that only about a half of the students cited technical problems, while the other half provided different arguments in favor of paper-based quizzes. Among non-technical reasons, students stated that they preferred seeing all of the questions at once, working on them in random order, and the ability to return to previously answered questions.

Due to the problems mentioned above, we decided to systematically explore the use of QuizPACK in a self-assessment context. Since Spring 2002, a range of QuizPACK individualized, self-assessment quizzes has been made available to all students of an introductory programming class. The quizzes were provided through the KnowledgeTree
portal [Brusilovsky 2004] along with other learning content such as lecture notes, program examples, and educational simulations. For each lecture of the course, the students were given access to one or two quizzes (of five parameterized questions each) that are focused on the programming constructs presented in that lecture. All work with QuizPACK was logged: each question attempted by a student left a record in a log file, complete with timestamp, student login name, and evaluation result. In addition, we have administered pre- and post-questionnaires each semester, to collect student demographic data (including their previous programming experience) and their feedback about the system. To motivate students to fill in a relatively long post-questionnaire, we offered three extra credit points for filling it in. However, since we were interested in an informed feedback, only students who attempted a specified number of questions were eligible to fill in this questionnaire.\(^1\) The log and questionnaire data were regularly used to evaluate the effectiveness of the systems and the students’ attitude towards it.

This paper summarizes the results of QuizPACK’s evaluation as a self-assessment tool in an introductory programming context for four consecutive semesters in 2002 and 2003. Since the context in which QuizPACK was used differed slightly between 2002 and 2003, we present the data separately for the two semesters of 2002 and the two semesters of 2003. In 2002, QuizPACK was used solely for delivering non-mandatory self-assessment quizzes. Though our evaluation results (see section 3.2) demonstrated that QuizPACK strongly benefited the students who used it, we discovered that the system was greatly underused. Less than a third of students used QuizPACK relatively actively - attempting more than 30 questions. Given the positive value of work with QuizPACK, we decided to provide some additional motivation for the students to use it by changing the format of classroom quizzes.

The role of classroom quizzes in motivating student learning is well known to teachers. In each of the four semesters in 2002 and 2003 we administered 8 to 10 classroom paper-and-pencil quizzes. During each quiz the students were allowed 10 minutes to answer 5 questions related to the topics of the two most recent lectures. In 2002 we used rather standard multiple-choice quizzes unrelated to QuizPACK. In 2003, we switched to fill-in-the-blank paper quizzes produced with QuizPACK using the following rule: For each of these quizzes, we selected five randomly instantiated QuizPACK questions out of the 15-20 questions offered for self-assessment on the

\(^1\) This arrangement did provide motivation to fill in the questionnaire - the majority of students who worked with the system enough to qualify did fill it in. However, as the data presented below shows, it didn’t provide enough additional motivation to use the system itself.
material of the two most recent lectures. The students were informed about this rule. Thus, the students who worked with QuizPACK after each lecture had a chance to practice each of the forthcoming classroom questions, though with different parameters. We hoped that this arrangement would motivate the students to use QuizPACK more broadly. Indeed, the percentage of students actively using QuizPACK increased more than 2.5 times: from 27% in 2002 to 70% during 2003. In addition, as the data presented below shows, this arrangement significantly changed the profile of student use of QuizPACK and increased the objective and subjective value of QuizPACK as a learning tool.

3.2 Objective Evaluation of QuizPACK Value

The goal of the objective summative evaluation of QuizPACK was to measure the objective value of the system as a learning tool. To determine the objective value we tried to find the relationships between students' work with the system recorded in the log file and their course performance. The student course performance had three major measures: (1) the total score on weekly in-class quizzes, (2) the final exam grade, and (3) the final course grade. Note that these parameters are quite different. In-class quizzes assessed students' knowledge of the syntax and semantics of C language. Because of the nature of QuizPACK, this knowledge was directly influenced by students' work with the system. In contrast, the final exam in our course measured mostly their programming skills: the ability to understand, modify, and write programs. The work with QuizPACK can influence these skills, indirectly, since they are based on an understanding of semantics. The final course grade (which was a combination of classroom quizzes, homework programming assignments and two exams) measured the full range of programming knowledge and skills.

Work with QuizPACK could be characterized by a number of measures. In our studies we used two relatively independent measures: activity and success. Activity was computed as the total number of QuizPACK questions attempted by the student. Success, which is the percentage of correctly answered questions, was calculated as the total number of correctly answered questions divided by the number of attempted questions. Note, that each parameterized question could be attempted several times. Each of these attempts (correct or incorrect) is counted in the activity and success measures. Most typically, students worked with the same question until the very first correct attempt; however, some of them kept working with the question even after the first success, which usually resulted in several registered successful attempts.
3.2.1 Using parameterized self-assessment quizzes without additional motivation.

During the two semesters of 2002, QuizPACK was available for 81 (39+42) students on the undergraduate level. Only 49 students tried QuizPACK, attempting an average of 38 questions. The number of students using QuizPACK “actively” (that is, attempting at least 30 questions) was much smaller, only 22 (see Table II for more data).

The profile of system use in 2002 was quite irregular (Figure 2). About 40% of the students merely tried the system on one or more occasions, attempting less than 35 questions (bar labeled “0” in the graph). Another 40% used it more regularly attempting between 35 and 71 questions (bar labeled “35” in the graph). The remaining 20% used the system quite seriously, yet there were large individual differences in the amount they worked with the system. The largest number of questions attempted by a single student over a semester was 319.

Given the irregular profile of system usage in 2002, it was hard to expect statistically significant connections between the students’ work with QuizPACK and their class performance. However, we were able to find some relationships, which gave insights into the pedagogical value of QuizPACK. To find the relationship between working with QuizPACK and class performance, we conducted regression analysis, a typical way to reveal cause and effect. As “explained” or dependent variables we used the students’ grade on the final exam, their in-class quiz performance and the final course grade (all measured in percents). Note that in-class quizzes used in the Fall of 2002 were regular multiple-choice quizzes, not a paper version of the QuizPACK quizzes. Explanatory variables were activity and success. Table I presents the results for the fall semester of 2002.

Table I. Results of regression analysis on class performance characteristics, versus QuizPACK use measures (Fall of 2002)

As we can see there is a strong, statistically significant relationship between the grade for in-class quizzes and student activity with QuizPACK: \( F(1,20)=13.37, \ p=0.0016 \). The coefficient of determination for this model is also
very high: $r^2=0.401$, which means that 40% of the grade for in-class quizzes can be explained by the activity variable. We also discovered the dependence between success and the final exam grade and between both activity and success and the final course grade. These two models do not show the same level of confidence, however; p-values for both of them are very close to the significance threshold value of 0.05. The values of the coefficient of determination for both models are also fairly high. Since in-class quizzes measure knowledge of semantics, we concluded that working with QuizPACK significantly increases the student’s knowledge of the semantics of the C language. We also discovered some evidence that working with the system contributes to the growth of higher-level knowledge and skills, which are measured by the final exam and course grade. As we wrote above, this data allowed us to regard QuizPACK as a strong learning tool and to experiment with increasing student motivation to use QuizPACK by switching to QuizPACK-based classroom quizzes in 2003.

3.2.2 Using parameterized self-assessment quizzes with additional motivation. During the Spring and Fall semesters of 2003, 73 students had access to QuizPACK, 60 of them tried it at least once, and 51 (70%) students worked with the system regularly. The average number of attempted questions was about 110.

Table II. QuizPACK usage in 2002 and 2003

As we can see from Table II, the use of paper-based QuizPACK quizzes for regular classroom assessment, changed the amount of system use dramatically. We measured the quantity of student work with QuizPACK using several measures. The average student activity (number of questions attempted) measures the plain volume of student work. The average number of sessions (number of independent sessions of work with the system) and the average course coverage (the ratio of questions attempted at least once, compared to the total number of questions) measure the distribution of this work over the duration of the course and course topics. The percentage of active students (students who attempted more than 30 questions) measures the distribution of this work over the students of the class. All these performance measures tripled or nearly tripled in 2003 in comparison with 2002 (it is actually surprising that the rates of growth for these different measures parallel each other).
The profile of student activity also changed considerably. Figure 3 shows both the very active use of QuizPACK by students and the even distribution of this activity among students. The maximum number of questions attempted by a single student rose to 510. At the same time, the average success score, measuring the quality of student work slightly decreased in 2003. We can suggest at least one reason for this: average course coverage tripled. The students regularly started to use more complicated quizzes (which have lower success rates) from the advanced sections of the course which had previously been ignored by most of the students in 2002.

To check the relationship between student work with QuizPACK and class performance in the new context, we performed a regression analysis similar to the one done with the 2002 data. In addition to the three class performance measures, we introduced a fourth: knowledge gain. We used this measure in hopes of finding a more reliable dependency between student work with QuizPACK and the growth of their knowledge of semantics. The problem is that in an introductory programming course, the beginning level of students’ knowledge of C and programming in general can vary significantly – from complete novices to students who are able to write reasonably-sized programs (the latter arguing that they needed this course as a refresher). It was natural to expect that the students’ performance measured at the end of the course depended not only on their work over the duration of the course (including QuizPACK work), but also on their past knowledge. To isolate the past experience factor we administered a pre-test (before the first lecture on C) and a post-test (during the Final exam). The paper-and-pencil pre-test and post-test featured the same ten QuizPACK fill-in-the-blank questions with different parameters. We calculated the knowledge gain, which measured the growth of student knowledge of semantics, as the difference between these post-test and pre-test scores.

Table III. Results of regression analysis on class performance characteristics versus QuizPACK usage (Spring and Fall of 2003)

Table III shows the results of the relationships analysis between QuizPACK work and course performance for two semesters of 2003. The grade for in-class quizzes and the final course grade were measured in percents, the final exam grade and the knowledge gain, in points. The new analysis confirmed the significant connection between QuizPACK work and the growth of students’ semantics knowledge. Both measures of
student knowledge of semantics - the knowledge gain and the grade for in-class quizzes - correlate significantly with the amount of work with QuizPACK. In addition, the larger number of actively working students in 2003 allowed us to discover significant relationships between QuizPACK the remaining two course performance measures. We found that the final exam grade correlates with the success variable. Students who strive to answer QuizPACK questions correctly have generally better grades on the final exam. The final course grade depends on both QuizPACK use measures. The p-values for all the models are well below the 0.05 threshold. Note that for models based on activity, we considered all 73 students in the class (from these, only 64 completed both the pre-test and post-test that we need in order to calculate the knowledge gain). For models based on success, we considered only the 60 students who had used QuizPACK at least once, since success can’t be calculated for the students who have not used the system.

3.2.3 QuizPACK as a learning tool in the context of programming classes. While our studies uncovered the relationship between usage of QuizPACK and class performance, this relationship does not imply the direction of dependency. Indeed, the most natural way to explain the relationship is to claim that the work with QuizPACK allows everyone to achieve better knowledge. It could be, however, that students in our course who already had a strong programming potential (due to a programming talent or a solid beginning level of knowledge) would have achieved better grades anyway, but simply preferred to use QuizPACK more. Fortunately, the specific organization of our course helped us to discover evidence in favor of the former scenario.

The introductory programming course that is the focus of our research consists of two components. The goal of the first component (6 lectures) is to introduce basic programming concepts (such as conditions and loops) with the help of Karel the Robot [Pattis et al. 1995], a simple, visual mini-language [Brusilovsky et al. 1997]. The goal of the second part (16-18 lectures) is to gain programming knowledge and skills while learning a reasonable subset of the C language. Both QuizPACK and paper-and-pencil quizzes were used for the second part of the course. While Karel and C share a solid number of programming concepts and constructs, they require different efforts to master them. Karel, due to its simplicity and visual programming environment, can be typically mastered relatively easily. Students with strong programming potential typically achieve very good results on the Karel part with little effort. In contrast, the C language has an abundance of second-level details, which require a significant effort to master them - even from students with strong programming potential. In the process of mastering the C language, QuizPACK may provide support that results in changed scores.
To determine whether QuizPACK has any pedagogical effect we compared student performance on the C parts of the course with their overall performance. We calculated class performance change, by dividing the grade for the C part (in percents) by the total course grade (in percents) that included both Karel and C performance. The grades for the Karel and C parts were compiled from grades for homework assignments and exam questions which focused on Karel and C respectively. The class performance change measures the relative change in performance during the C part of the course as compared with the whole course (Karel plus C). The class performance change is less than 1 for students who achieved better results in the Karel part of the course and greater then 1 for students who achieved better results in the C part. The larger this proportional change is, the better the relative grade increase for the C part. If the work with QuizPACK can help students to improve their “native” programming performance, as measured by the Karel grade, we may expect a positive correlation between class performance change and the amount of work with the system. If the work with QuizPACK harms student progress, we may expect a negative correlation. If the student performance is mainly determined by their programming potential and other aspects not related to QuizPACK, we may expect no correlation between these parameters (and we can expect no visible difference between the C part and whole-class performances). While any of these conclusions still cannot be considered an ultimate proof, the class performance change analysis provides more reliable evidence about the role of QuizPACK than simple correlation analysis.

Table IV demonstrates the results of regression analysis on class performance change versus activity. There is strong statistical evidence that change in course performance is connected to usage of QuizPACK: F(1, 43)=7.79, p=0.0078. The value of the coefficient of determination tells us that 15.3% of the change in the students’ grade for the C part of the course can be explained by the activity of their work with self-assessment quizzes. This data gives as better evidence to conclude that working with QuizPACK has caused the observed growth of the students’ knowledge. If we hypothesize that student performance on the Karel part reflects their programming potential, then work with QuizPACK allows the students to leverage their programming potential and even reach beyond it. Students who have not invested an effort to learn the details of C with the help of QuizPACK may expect their course performance to drop below their potential. In contrast, the students who worked with QuizPACK a lot may expect to perform above their starting potential.
In Fall 2003, we used another approach to help determine the role of QuizPACK as a learning tool. During that semester, the School offered two sections of the Introduction to Programming course, naturally forming two groups: experimental (28 students) and control (22 students). The syllabus and class settings for both classes were comparable. The students in the experimental group had access to QuizPACK during the course (and were motivated by their classroom quizzes to use it), whereas students in the control group had no access to the system. At the beginning and end of the course, each group took the same pre- and post-tests. The results of these tests were used to calculate knowledge gain. The average knowledge gain for the control group was 1.94 (Sigma = 1.55); for the experimental group it was 5.37 (Sigma = 2.17). In other words, motivated use of QuizPACK more than doubled the knowledge, providing a difference that is larger than 1.5 standard deviations. Such a result is typically considered to be very strong evidence in favor of an educational technology [Bloom 1984].

In addition to determining the value of QuizPACK to the class as a whole, we attempted to determine in our research whether QuizPACK provides different impacts on different categories of students. We were interested in finding the categories of students who benefit more or less from the system than others. As categorical variables we attempted to divide the group by such students’ characteristics as gender, domain knowledge (measured by final letter grade), initial programming experience, and even learning style. Unfortunately, none of these variables were able to divide students into categories that were differently influenced by QuizPACK usage. We tried both simple classification and clustering approaches. To reveal possibly hidden groups of students formed by combinations of categorical variables, we applied the expectation-maximization algorithm for Naïve Bayes with a hidden class variable and missing values. This algorithm has many desirable features: a strong statistical basis, theoretical guarantees about optimality, easily explainable results, and robustness to noise and to highly skewed data. [Dempster et al. 1977]. The cluster picture we received did not have intuitive interpretation and did not match the data from semesters not used for model training.
3.3 Subjective Evaluation of QuizPACK Value

To evaluate the students’ subjective attitudes toward the system, we administered a questionnaire at the end of each semester to each class using the system. The questionnaire included a range of questions focused on the system’s value, its features, and aspects of its usage. The number of questions grew from 9 in Spring 2002 to 14 in Fall 2003. The participation was voluntary, however, to motivate students to participate in the questionnaire, we offered three extra credit points to every qualified participant. A student was qualified to take the questionnaire if he or she worked with at least 10 or more 5-question quizzes, associated with 6 or more different lectures, and used this feature during several sessions for at least 20 days, before being offered the chance to take the questionnaire. In total, 101 filled questionnaires were collected for the four semesters of 2002 and 2003.

In the following analysis we focus on the six multiple-choice questions that are most relevant to the topic of this paper. The questions, the possible answers, and the number of students that selected each of these answers are listed in Table V. Each of these questions (as well as the majority of questions in the questionnaire) provided a student with four optional answers: strongly positive, positive, neutral, or negative. These choices were coded by values from 3 (strongly positive) to 0 (negative). The profile of student answers is summarized in Figure 4. This chart represents the feedback of all 101 students over the 4 semesters (with the exception of question 6 which was not used in the Spring semester of 2002).

Table V. Six questions about QuizPACK and the summary of student answers

As this analysis shows, the students were very positive about the system. On average, from 80% to 90% of the students provided at least light-positive answers to the questions. More than 50% answered that the system significantly helped them during the course. More than 40% considered the system as a critical part of the course and were ready to recommend it strongly to other students taking this course. We have been evaluating different systems in the classroom and this is the strongest result we have ever achieved. QuizPACK was the clear champion of our class. When comparing all support tools they used during the class, students rated QuizPACK as the second most valuable, right after
lecture slides (in Spring 2002 they actually preferred QuizPACK). Even the relatively
plain system interface – arguably the least-appreciated feature of the system - received
more than 80% light-positive feedback.

We also noticed that the profile of student answers to the question 2 about the
system's ability to generate the same question with different data is close semester-by-
semester to the profile of their answer to the question 1 about the helpfulness of
QuizPACK. We hypothesized that the students’ highly positive attitude to QuizPACK is
strongly connected with the ability of the system to deliver and evaluate parameterized
quizzes. Statistical analysis confirms this finding. The correlation coefficient between the
general attitude to the system and their attitude toward question parameterization is fairly
high – 0.48. It is larger than corresponding correlation coefficients for any other system
feature.

The regression analysis on students' attitude to QuizPACK in general against their
attitude toward different features of the system supports these results (see table VI).
Students' opinions about three main features (parameterized question generation, type and
content of question material, and system interface) have a statistically significant
relationship to the evaluation of the system in general. Question generation has both the
strongest relationship ($F(1,99)=29.52, \ p=3.98E-7$) and the largest coefficient of
confidence ($r^2=0.23$) among all the features. However, the most interesting model
seems to be the last one, which regresses the general attitude to the system against all
three partial attitudes. This model has even larger values of confidence ($p=1.48E-8$)
and coefficient of determination ($r^2=0.332$). It means that all three features contributed
significantly to the overall students' attitude toward the system.

Table VI. Results of regression analysis on students' attitude to QuizPACK in general
versus attitude toward different system features (spring and fall of 2003)

To complement the general analysis, we attempted to compare the attitude toward
QuizPACK among different groups of students. We determined three characteristics for
dividing students into groups: gender, final course grade and initial programming
experience (novice, some experience, considerable experience). To compare the attitude
of different groups of students to the system and its features, we calculated an average
attitude for each of the compared groups for every question and every semester by
averaging the answer codes (Table V). Then we compared the profile of their answers,
split by each feature. We were not able to find any consistent difference between groups
of students with different initial experience. At the same time, students of different genders as well as those with different final course grades did demonstrate distinct attitude patterns that were repeated semester by semester.

Figure 5 demonstrates the difference between profiles of answers given by female students and male students to the selected set of questions. The average answers in this figure are shown in percentage of maximum positive answer (i.e., 100% means that all students in this category give the strong-positive answer). It is remarkable that the graph, corresponding to female students, dominates the graph for male students for almost all questions (except the question about parameterized generation). The difference is quite visible; it varies from 3% to almost 10% for several questions. While Figure 5 shows the summative comparison over 4 semesters, we observed the same dominance every semester. Females’ more positive attitude correlates with the fact, that, on average, female students were more active with QuizPACK, (For students who participated in the questionnaire, the average value of activity for female and male students were about 130 and 98, respectively.) At the same time, the difference between average values of success for female and male students was almost zero (51.78% of correct answers for female students and 50.66% for male students). We may hypothesize that the high attitude of female students towards QuizPACK and the higher activity of their work with the system express the fact that QuizPACK was a more critical tool for female students over the duration of the course. It is remarkable that in our course, female students, who usually lag behind male students in programming subjects, achieved the same average grade as male students.

Figure 6 demonstrates the similar influence of student knowledge of the subject, as measured by the final grade. One of the lines on the graph shows the average answers of students who got positive final course grades (from B- to A+); another shows average answers of the rest of the class. As we see from the figure, the AB line is above the C&less line for all questions but one. For two questions, the difference is about 15%, for three others it is close to 5%. The difference in QuizPACK activity for these groups of students is very similar to the difference in attitude. Average value of activity for AB-
students is about 120, while for CDF-students it is only 47. The average values of success measure for these two groups are nearly the same (51.07% and 50.52%).

Fig. 7. Change of students’ attitude toward the system after the introduction of QuizPACK-based classroom quizzes

Figure 7 demonstrates the change in attitude toward the system after the introduction of QuizPACK-based paper quizzes. As we can see, in 2003, students were much more positive about QuizPACK than a year before. Not surprisingly, their attitude toward the type and content of questions that were mostly the same in 2002 and 2003 changed very moderately. However, their appreciation of the parameterized question generation that became more important in the 2003 system went up dramatically from 70% to 90%. Similarly, their overall attitude toward the system, such as the evaluation of its helpfulness or its role in the course rose about 20%.

The differences presented above may have two explanations. First, we can argue that the system is simply more helpful for the groups of students who demonstrated better attitudes toward the system (i.e., females, students with better knowledge of the subject and students who are motivated by QuizPACK-based paper quizzes). The same reason caused the increased use of the system by this group. However, the opposite may also be true: increased use of the system is the reason for a better attitude. Further studies are required to compare the value of the system for students of different gender and with different level of knowledge of the subject.

3.4 Assessment or Self-Assessment?
As we have pointed out above, our early attempts to use QuizPACK for classroom assessment caused generally negative students’ reactions. To solicit students’ opinion about the most appropriate context for using the system, we added a focused question in every questionnaire administered in 2002 and 2003. This multiple-answer question asked students to check all contexts where they would like to use QuizPACK (Figure 8). As we can see, the option “right in class in assessment mode to replace the paper quiz” was the least popular. Only 23 students (about 10% of the total number of answers) selected it while the remaining 90% selected exclusively self-assessment contexts. Among scenarios of using QuizPACK for self-assessment, the currently suggested out-of-class self-assessment was the champion.

Fig. 8. Students’ preferences for the context of using QuizPACK (summary statistics over 2002 and 2003)
In 2003, we added two additional questions to the questionnaire, to solicit student feedback about the new arrangement for in-class quizzes. The analysis of the answers for the Spring and Fall semesters of 2003 showed that students mostly approved of this innovation. When answering the question about use of QuizPACK-based paper-and-pencil quizzes for classroom assessment, 66.67% of the students wrote that “it was a very good arrangement,” 25.00% believed, that “it was quite good,” 8.33% answered “it made some sense, though it was far from perfect,” and none of the students gave the answer “it was a completely wrong arrangement.” Assessing the value of QuizPACK as a tool to help them prepare for classroom quizzes, 64.58% answered that “it helped them a lot,” 27.08% that “it was quite helpful,” 8.33% that “it helped a little,” and no one answered that “it did not help at all.” Thus about 2/3 of students provided strong positive feedback about the new arrangement with more than 90% of the total students being positive or strongly positive.

We think that the student answers to the three questions analyzed above provide good evidence that our most recent arrangement is one of the most attractive and useful ways for students to use the parameterized code evaluation quizzes. This arrangement provides an interesting combination of assessment and self-assessment: using QuizPACK to deliver automatically-graded self-assessment quizzes in an out-of-class context while assessing students in the classroom through paper-and-pencil quizzes that have been generated with QuizPACK.

4. QUIZPACK AND THE AUTOMATED EVALUATION OF PROGRAMMING KNOWLEDGE

This paper is not complete without an attempt to place QuizPACK into the field of other work on automated evaluation of programming knowledge. There are several ways to classify the range of existing systems that partially cover this special issue. From the authors’ prospect, the most important difference is the kind of knowledge that a specific system is trying to assess. In the past, it was customary to distinguish between three kinds of knowledge to be taught in a programming course: knowledge of syntax (how to compose valid programs), semantics (how programming constructs work), and pragmatics (how to solve typical problems). All three kinds of knowledge are typically evaluated in various tests and assignments. During the last 15 years, researchers in the area of automated evaluation of programming knowledge focused mainly on the knowledge of semantics and pragmatics (the syntax knowledge are considered much less important nowadays than in the early days of computer science education).
Correspondingly, two kinds of automated assessment systems emerged: program evaluation and program generation. Systems based on program evaluation attempted to test students’ semantics knowledge by asking them to trace (evaluate) an existing program and then by assessing the results of this tracing. Systems based on program writing attempted to test students’ pragmatics knowledge by asking them to write (generate) code and then by assessing the produced code.

Since the ability to solve programming problems and produce working code is considered most important for Computer Science graduates, the majority of work on automated evaluation of programming knowledge was devoted to the latter kind of systems. A number of program assessment systems on different scales were produced over the last 15 years. From the knowledge assessment point of view, these systems differ in two aspects: the amount of code they require the student to write and the level of code analysis. Question-oriented systems typically require the student to fill a relatively small gap in an existing program [Arnow & Barshay 1999; Garner 2002]. Assignment-oriented systems assess complete programming assignments, often written from scratch or from a small program skeleton [Benford et al. 1994; Daly 1999; Higgins et al. 2006; Higgins et al. 2003; Jackson & Usher 1997; Joy et al. 2006]. While the technology behind these systems is often similar, the question-oriented systems assess mostly student abilities to use programming constructs in context and the assignment-oriented systems assess mostly the ability to solve programming problems. These skills belong to different level of Bloom’s Taxonomy of Educational Objectives [Bloom 1956].

The QuizPACK system presented in this paper belongs to the first class of systems based on program evaluation (tracing). These systems are currently less popular among developers. In addition to QuizPACK, we can only cite a range of tutors developed by Amruth Kumar [Dancik & Kumar 2003; Fernandes & Kumar 2002; Kostadinov & Kumar 2003; Kumar 2006; Kumar 2002; Kumar 2005]. The evaluation-based approach, however, is a popular way of assessing student knowledge in the neighboring area of algorithms and data structures. Such systems as PILOT [Bridgeman et al. 2000], TRAKLA [Korhonen & Malmi 2000], TRAKLA2 [Malmi et al. 2006], and MA&DA [Krebs et al. 2005] are similar to QuizPACK and Kumar’s tutors: they ask a student to mentally execute an algorithm from a specific starting state and submit a trace of actions or the final results of algorithm execution. We can hope that all these cited works will draw more attention to the importance of assessing students’ knowledge of semantics and their program-tracing skills. We think that this knowledge is an important component of the student’s ability to understand and write programs. This opinion is shared by a large
team of researchers [Lister et al. 2004], which has recently stressed the importance of assessing student program-tracing skills.

5. SUMMARY AND FUTURE WORK
This paper explores a specific kind of educational activity in the field of programming: parameterized code-execution exercises. In a code-execution question, the student is provided with a fragment of a program and is asked to predict the value of a particular variable or a string to be printed after execution of this fragment. We suggested an approach to automatic generation and evaluation of code-execution exercises. The approach is based on runtime execution of parameterized code fragments. We also developed the QuizPACK system, which implements the suggested approach for the specific case of teaching the C language. The focus of this paper is the classroom evaluation of the QuizPACK system over four consecutive semesters in an introductory programming classroom.

Our studies demonstrated that QuizPACK, used in an out-of-class self-assessment mode, is an exceptional learning tool. As demonstrated by regression analysis and comparison with a control group, the work with QuizPACK significantly improved students’ knowledge of semantics and positively affected higher-level knowledge and skills. The students themselves praised the system highly as a helpful learning tool. About 90% of the students stated that the system helped or significantly helped them during the course and that they recommend it to their friends taking the same course. The ability to generate parameterized exercises was rated especially high among system features and, as shown by correlation analysis, contributed to their overall satisfaction with the system.

We also demonstrated that the use of the system in self-assessment mode can be significantly increased (with a corresponding increase in positive student attitude and knowledge gain) by motivating students through QuizPACK-based classroom paper-and-pencil quizzes. While we originally planned to use QuizPACK for automating knowledge evaluation in the classroom, our data indicates that a combination of QuizPACK-based classroom paper-and-pencil quizzes and out-of-class Web-based self-assessment with QuizPACK is among the most student-preferred ways of using the system.

Our current work is focused on providing adaptive guidance for the users of QuizPACK in order to help each student select the most useful exercises, according to their learning goals and level of knowledge. Our first experiments with adaptive guidance demonstrated that in addition to guiding the students to the most relevant exercises, it
also further increased student motivation to work with the system, resulting in a further improvement of student knowledge [Brusilovsky & Sosnovsky 2005]. We plan a more comprehensive exploration of this phenomenon. We also hope that the QuizPACK system will begin to be used more broadly, enabling us to collect more data and perform some deeper analyses of QuizPACK’s influence on different categories of students. In particular, we intend to explore QuizPACK with new categories of students, such as high school students and community college students.

REFERENCES


DALY, C. 1999 RoboProf and an introductory computer programming course, SIGCSE Bulletin - Inroads. 31, 3, 155-158.


JACKSON, D., USHER, M. 1997 Grading student programs using ASSYST, SIGCSE bull. 29, 1, 335-339.

JOY, M., GRIFFITHS, N., BOYATT, R. 2006 The BOSS online submission and assessment system, ACM Journal on Educational Resources in Computing. This volume.


