USING A JAVA CERTIFICATION BOOK AND MOCK EXAM IN AN INTRODUCTORY PROGRAMMING COURSE

John K. Estell

Electrical & Computer Engineering and Computer Science Department

Ohio Northern University

Ada, OH 45810

Introduction

Introductory programming courses, such as those taught in Java, provide foundations upon which disciplines such as computer science and computer engineering are structured. Accordingly, it is appropriate to employ mechanisms that determine whether the student is adequately prepared and if the program provides sufficient coverage to ensure that the foundations are solid. Certification exams are one way for Java programmers to validate their skills through use of an externally designed standard. It is common for a programmer to prepare for such an exam by purchasing a certification guide and practice by working through review questions and taking mock exams. Java certification guides allow one to review the essential features of the Java programming language, their syntax, and their correct usage. While the typical Java certification guide does not serve as a language reference nor does it promote particular programming techniques, it does serve as a primer on selected topics where it has been determined that a need for mastery exists. Additionally, these guides contain a repository of questions tailored to examine whether or not someone had achieved such mastery. So why not use this material in the evaluation of this portion of the undergraduate curriculum?

Implementation

This paper examines the use of a Java certification guidebook in a third quarter introductory programming course; the students have previously had two quarters of instruction in the C++ programming language. The guidebook, "A Programmer's Guide to Java Certification," by Mughal and Rasmussen,[1] was used in conjunction with a traditional textbook to provide a complete treatment over the course material. This guidebook was written to provide coverage over the certification objectives of the Sun Certified Programmer for the Java 2 Platform (SCPJ2) 1.4 Exam. However, the preface to the book makes it clear that it is not meant to be a complete Java reference, nor is it meant to be a book on teaching one how to program. Instead, it emphasizes the salient features of the Java programming language, their syntax, and their correct use. The areas covered include language fundamentals, operators and assignments, declarations and access control, control flow, exception handling, object-oriented programming, nested classes and interfaces, object lifetime, threads, fundamental classes, and the Collections framework. Each chapter in the book contains a discussion of the pertinent concepts being covered, example code, programming exercises, and sets of review questions. The review questions are in multiple choice format, with the number of correct answers for each question being provided. One of the appendices contains the annotated solutions to the review questions, featuring either a brief explanation as to why the answer is correct or why the other responses are incorrect. The book also provides information regarding taking the SCPJ2 1.4 Exam, study notes over the exam objectives, and a mock exam with separate, annotated answers.

The course was organized such that students were assigned periodic readings from both the certification book and the textbook; the rate of coverage was roughly one chapter from the certification book per week. At the end of each week the students were given a quiz based on a subset of the review questions contained within

that week's reading in the certification guide. At the end of the course, a mock certification exam similar to the SCPJ2 1.4 exam was given to the students as their final exam in the course. The final exam contained 61 multiple choice questions similar to those one would expect to appear on an actual certification exam and was conducted in a two hour exam session, the same amount of time given to those individuals taking the SCPJ2 1.4 exam.

Rationalization

There are multiple reasons as to why it is thought beneficial to incorporate the use of certification books and mock exams into courses. First, there are already programs that have incorporated preparation for Java certification into their curricula. The predominant rationalization provided was that of marketability. Ortiz's paper[2] contains a discussion on the relevance and value of information technology (IT) certifications in general, and to the balancing act needed in the academic environment between training and educating students in preparation for their careers. The paper by Al-Rawi, Lansari and Bouslama[3] focuses on the challenge of keeping current one's curriculum and how the integration of certification objectives provided their students with a sufficient body of knowledge to take the SCPJ2 1.4 exam; it was reported that two-thirds of their students successfully passed. Other programs have looked at the integration of certification training into an academic program; in his paper, Zeng[4] discusses a curriculum model in place at Indiana Wesleyan University for IT certifications in a two-year associates degree program and tackles the larger discussion of the necessity of professional certification. One of his arguments in support of his position is that certification provides credibility, but he also notes that it does not provide the same level of credibility at that obtained through licensure or accreditation.

Through the use of certification material one raises the issues of professionalism and life-long learning. Those who go into one of the IT fields usually find themselves in an ever-changing environment, as the languages and technologies experienced as an undergraduate quickly become obsolete. As a consequence, in order to survive and prosper in the corporate environment, one must be able to learn new languages and technologies essentially on their own. Exposure to certification books provides an insight to a systematic approach to learning, especially in the context of requiring a student to adapt to a different language because of the nature of the introductory programming sequence at the author's institution. This provides students with a practical, real-world skill not commonly found in a computer science or engineering curriculum. Additionally, certification preparation often delves into some of the more esoteric areas of a language; one is forced to develop a more detailed knowledge of the material in order to deal with these aspects, and that in and of itself is a benefit as it leads to a greater understanding of the language. The concept of certification definitely hits upon the attributes of what one would expect of an IT professional, as certification serves as proof that the recipient has achieved a certain level of competence in that particular field. In context to this paper, to become a Sun Certified Programmer one must take the SCPJ2 exam. This exam consists of 61 primarily multiple choice questions; there are occasionally a few short answer-type questions. In order to obtain certification, the candidate must score 32 or more correct answers; this constitutes a lower bound of 52% for passing the exam. While 52% sounds like an easy goal, experience has proved otherwise. One observation published in Certification Magazine[5] noted that prospective candidates spend up to six months in preparation for the exam, depending upon the candidate's background, with those possessing one or more years of recent Java experience requiring only three to 12 weeks of preparation. Part of the difficulty is that some of the questions can be deemed esoteric or that it requires one to become a human complier; however, it is these sorts of questions that separate the code slingers and the "video IDE players" from programmers who understand the

nature of the beast that they are dealing with. By exposing students to these types of questions within such an environment, one raises the level of expectations regarding their knowledge and understanding of the material.

The final area of interest that benefits from this pedagogy is that of assessment. Assessment, when properly used, allows an institution to evaluate the effectiveness of a curriculum and, when combined with continuous quality improvement practices, provides a mechanism to identify and improve weaknesses in the curriculum. The use of certification preparation materials is attractive as they provide a sizeable bank of already categorized questions which can be used for assessment purposes. Additionally, the use of a mock SCPJ2 certification exam provides a quasi-external performance validation; a program can be said to be doing well in schooling their students on the fundamentals of Java programming if a sizeable portion of their students can "pass" the mock exam at a rate equal to or higher than the established rate for the actual certification exam. In fact, a mock exam possesses a major benefit over using the actual certification exam in that, for the actual exam, only the final score is given; no information is available as to what questions were missed. In a classroom-administered mock exam, the instructor has full access to all exams, and so can categorize all problems into their appropriate objective areas. Following the grading of the exams, the individual problem scores can be collectively analyzed and the performance in each objective area evaluated; this allows for targeted action plans to be implemented if poor performance is noted in a specific objective, thereby improving the curriculum.

Results

Ten quizzes containing a total of 77 questions and covering 10 chapters in the Mughal and Rasmussen book were given to the students on a weekly basis during the quarter; an example quiz is presented in the Appendix to this paper. As an incentive to review the material, the questions used for the quizzes were taken directly from the review questions in the book; however, various modifications were made (in terms of the ordering of the answers and subtle changes made occasionally to the code and/or text that would change the answer from that given in the book) in order to ensure that the quizzes could not be passed simply through rote memorization of the answers. As the quizzes ranged from five to nine questions, and the number of review questions for each chapter ranged from 11 to 33, this resulted in students being strongly encouraged to at least look at all of the review questions and, hopefully, to work the review questions to prove to themselves that they possess a sufficient mastery of the material. As all of the quiz questions were in the multiple choice format, it was relatively simple to create a spreadsheet where each student's incorrect answers could be recorded on a question by question basis. Not only did this allow for an accurate representation of individual student performance, it allowed for easy analysis of group performance with respect to both individual questions and by the specified objectives. The questions were collected into their respective exam objective categories and analyzed through use of 4-tuple performance criteria vectors.[6] Performance criteria vectors are constructed with data from only those students who received a passing grade in the course, as the primary question is whether students are passing the course without achieving the specified course outcomes. The "exemplary" categorization was used for students getting at least 85% of the questions correct (which, given the limited number of questions in each category, meant that the student missed at most just one question). The "adequate" category described a performance level where students got between 70% and 85% of the questions correct. The boundary between "minimal" and "unsatisfactory" was set at 52%, which is the established minimum for passing the SCPJ2 exam. Table 1 shows the performance criteria vectors for the ten measured certification objective areas.

Table 1. Quiz results classified by certification objectives.

	Exemplary	Adequate	Minimal	Unsatisfactory
Basics of Java Programming	22	6	2	0
Language Fundamentals	17	6	4	3
Operators and Assignments	13	10	5	2
Declarations and Access Control	11	9	3	7
Control Flow and Exception Handling	8	15	3	4
Object-Oriented Programming	11	11	5	4
Object Lifetime	20	5	3	2
Threads	19	8	1	2
Fundamental Classes	18	5	6	1
Collections Framework	6	19	1	4

An examination of the certification objectives data indicates reasonable performance overall; however, some red flags were raised based on higher than expected values in the minimal and unsatisfactory categories. For example, over 20% of the class performed at an unsatisfactory level on the declarations and access control objective. An investigation showed that the students performed poorly on questions concerning arrays, and it was eventually discovered that the amount of array coverage in the prior courses was less than what the instructor of this course assumed was being covered. Subsequent discussion resulted in an action plan that called for increased coverage of arrays in the prerequisite courses and for additional emphasis on arrays in this course.

A mock Java certification exam consisting of 61 multiple choice questions was used for the final exam in this course. Final exams are scheduled in two-hour blocks of time at the author's institution, which coincidentally is the same amount of time allotted for taking the SCPJ2 exam. The overall results of the exam are shown in Figure 1; the horizontal line indicates the boundary between passing and failing if this were an actual certification exam. On the final exam, 17 out of the 30 students scored at least 32 questions correct for a passing rate of 57%, which is not too far off from the actual certification exam performance reported by Al-Rawi, Lansari and Bouslama.[3] There are two items that are worth noting regarding this exam. First, there was a cluster of students – nine in all – having scores within five points of passing; second, the average amount of time spent on this exam was only 69 minutes (the author has a habit of recording the elapsed time on the front page of each student's exam when that exam is turned in). In particular, the average amount of time spent by the clustered students on the exam was 65 minutes, with 83 minutes being the longest amount of time spent within this group. Given that the length of the exam was 120 minutes, all of these students had sufficient time to review their work but decided not to do so. Had members of this group reviewed their work and corrected at least five of their errors, the class pass rate would have jumped up to 87%. This data indicates that the students, at least in this cohort, could benefit from information regarding test-taking strategies, including the benefits of reviewing one's work when time permits. In terms of the accuracy of the mock examination, one student from the class took the actual SCPJ2 1.4 exam approximately three weeks after the final exam. The student received a passing score of 52 out of 61 questions (85%) answered correctly, compared to his final exam score of 49 out of 61 questions (80%) answered correctly. While one external datum does not provide statistically convincing proof, it does provide at least a modicum of verification for the use of the mock certification exam in this manner.
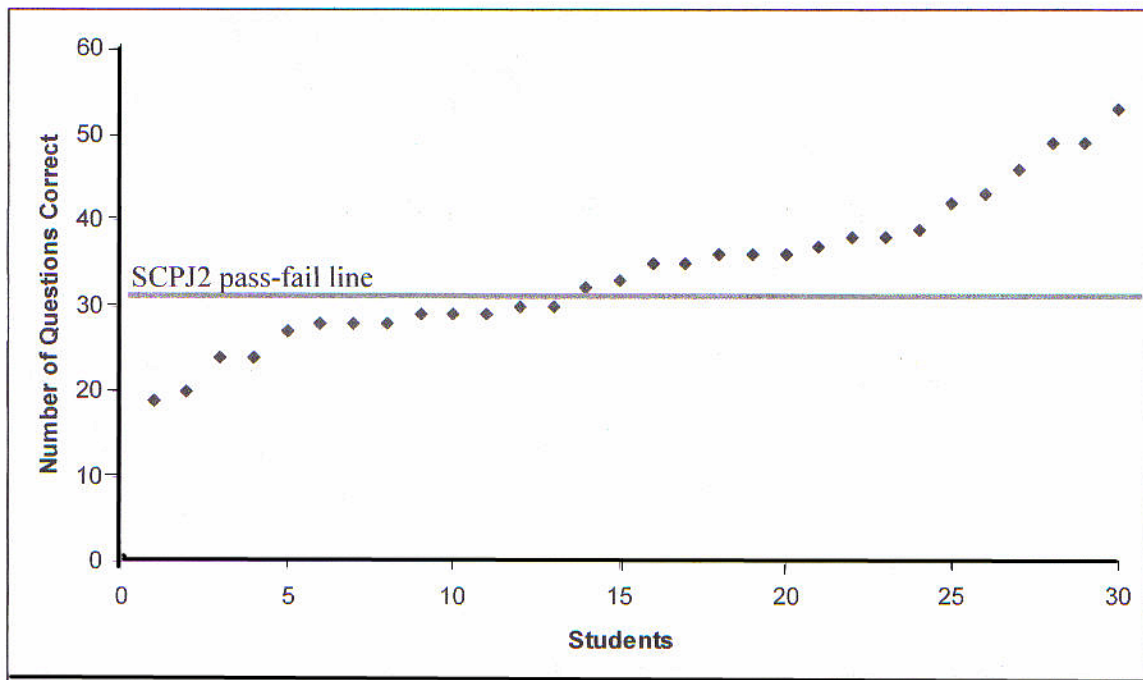
Figure 1. Results from final exam.

The results of the mock certification exam were also used as a component for the department's program outcomes assessment process. Performance criteria vectors were again used for the evaluation, with the same boundary points being employed to delineate the four categories. Data from this exam was extracted for only three components of our program outcomes: understanding of object-oriented programming, the ability to examine and evaluate code, and language fundamentals. The results, shown in Table 2, are not as promising, from the point of view of class performance, as those obtained from the quizzes, but are of great value for the purpose of assessment. First, it should be noted that, with a relatively small sample set of questions for these categories on the exam, the data looks worse than it probably should, given the performance on the quizzes. Concerning the results in the object-oriented programming area, many of the final exam questions were not Java-specific. Collectively, students did perform well on these general questions, but had difficulty with some of the Java-specific questions, which lowered the overall performance. Part of the difficulty in this area appeared to be from the differences in coding and implementation approaches between C++ and Java. With the code evaluation questions, the sub-par results required an

Table 2. Final exam results classified by program outcomes.

	Exemplary	Adequate	Minimal	Unsatisfactory
Object-Oriented Programming	5	8	9	8
Code Evaluation	2	9	10	9
Language Fundamentals	4	3	9	14

inspection of the individual questions. On some of the problems the class performed very well whereas on others there were considerable difficulties; part of this was due to not understanding some of the more "interesting" features of a C-based language, such as how in a switch statement the program flow of control starts at the matching case statement and continues onward from there. As an example, only 32% of the students recognized the need for, or understood the purpose of, a break statement within the body of a switch statement for one particular program provided as an exam question. The questions that comprised the language fundamentals category were particularly nasty, with several testing knowledge in some of the darker corners of the language specification, so it is not surprising that almost half of the class did not perform at a satisfactory level. Overall, the analysis of the data required only a short amount of time and has resulted in the identification of areas where improvements can be made in the curriculum.

Conclusion

Ideally, the certification guidebook and mock exam would be used at the end of a one-year course sequence conducted entirely in the Java programming language. This was not the case here, as the previous two courses were conducted using a different language. It is a lot to ask out of a freshman to learn a new, albeit closely related, language while studying a certification guide for that language at the same time; this is probably why the one major area of concern expressed by the students in their course evaluations was that the amount of reading material was considered to be high.

The individual scores on the quizzes and the exam serve as feedback to the student as to whether sufficient mastery of the material has been achieved. The performance criteria vectors for use in course and program outcomes assessment can be easily generated from these scores such that relevant assessment information can be extracted; that information in turn is used to implement improvements in the curriculum.

These results have shown the use of both the Java certification guidebook and the mock certification exam to be a successful methodology.

Bibliography

1. K. A. Mughal, R. W. Rasmussen, A Programmer's Guide to Java Certification, 2nd ed., Addison-Wesley, Boston, 2004.

2. A. Ortiz, Preparing Undergraduate Students for Java Certification, OOPSLA '03 Proceedings, 2003.

3. A. Al-Rawi, A. Lansari, F. Bouslama, Integrating Sun Certification Objectives into an IS Programming Course, Informing Science and Information Technology Education Joint Conference, Flagstaff, Arizona, June 2005. Available online:http://2005papers.iisit.org/I20f39Rawi.pdf

4. F. F. Zeng, A New Approach to Integrate Computer Technology Certifications into Computer Information System Programs, Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition.

5. D. L. Hecksel, M. Green, Java Certification: Who, What, When, Where, Why and How, Certification Magazine, July, 2000. Available online:http://www.certmag.com/issues/jul00/feature_hecksel-green.cfm.

6. J. K. Estell, "Streamlining the Assessment Process with the Faculty Course Assessment Report," Proceedings of the Best Assessment Practices VIII Symposium, Rose-Hulman Institute of Technology, Terre Haute, IN, 2006.

Biographical Information

John K. Estell is Chair of the Elecrical & Computer Engineering and Computer Science Department at Ohio Northern University. He received his doctorate from the University of Illinois at Urbana-Champaign. His areas of research include simplifying the outcomes assessment process, user interface design, and the pedagogical aspects of writing computer games. Dr. Estell is a Senior Member of IEEE, and a member of ACM, ASEE, Tau Beta Pi, Eta Kappa Nu, and Upsilon Pi Epsilon.

Appendix: Example Quiz

Spring Quarter 2005 – Quiz 4 – April 1, 2005   Section: 10 am   11 am

1. Given the following pairs of method declarations, which statements are true? Select the two correct answers.

```
void fly(int time) {}
void Fly(int time) {}

void fall(int distance) {}
int fall(int time, int speed) { return time*speed; }

void glide(int time) {}
int  glide(int distance) { return distance; }
```

    a.   The first pair of methods will not compile correctly.
    b.   The second pair of methods will not compile correctly.
    c.   The third pair of methods will not compile correctly.
    d.   The first pair of methods will compile correctly and overload the method name fly.
    e.   The second pair of methods will compile correctly and overload the method name fall.
    f.   The third pair of methods will compile correctly and overload the method name glide.

2. Given a class named Card, which one of the following is a valid constructor declaration for the class? Select the one correct answer.

    a.   `public static void Card( String[] args ) {}`
    b.   `void Card() {}`
    c.   `private final Card() {}`
    d.   `Card Card() {}`
    e.   `Card( Card c ) {}`
    f.   `abstract Card() {}`

3. Which of the following statements are false? Select the three correct answers.

    a.   A constructor can be declared private.
    b.   A constructor can return a value.
    c.   A constructor must initialize all the fields of a class.
    d.   A constructor can access the non-static members of a class.
    e.   A constructor can be overloaded.
    f.   All classes must define a constructor.

4. Given the following member declarations, which statement is true? Select the one correct answer.

```
int a;                      // (1)
static int a;               // (2)
int f() { return a; }       // (3)
static int f() { return a; } // (4)
```

    a.   Declarations (1) and (3) cannot occur in the same class definition.
    b.   Declarations (2) and (3) cannot occur in the same class definition.
    c.   Declarations (1) and (4) cannot occur in the same class definition.
    d.   Declarations (2) and (4) cannot occur in the same class definition.

5. Given the following definition of a class, which fields are accessible from outside the specified package? Select the two correct answers.

```
package edu.         ;

public class Foo {
    int a;
    private int b;
    protected int c;
    public int d;
}
```

    a. Field b is accessible in subclasses only in other packages.
    b. Field c is accessible in subclasses only in other packages.
    c. Field d is accessible in subclasses only in other packages.
    d. Field a is accessible in all classes in other packages.
    e. Field b is accessible in all classes in other packages.
    f. Field c is accessible in all classes in other packages.
    g. Field d is accessible in all classes in other packages.

6. Which statements are false? Select the **two** correct answers.

    a. If a and b are of type boolean, the expression (a = b) can be the conditional expression of an if statement.
    b. An if statement can have either an if clause or an else clause.
    c. The statement if (false) ; else ; is illegal.
    d. Only expressions which evaluate to a boolean value can be used as the condition in an if statement.
    e. The conditional expression of an if statement can have method calls.

7. Which of these combinations of switch expression types and case label value types are legal within a switch statement? Select the one correct answer.

    a. switch expression of type float and case label value of type int.
    b. switch expression of type byte and case label value of type float.
    c. switch expression of type char and case label value of type long.
    d. switch expression of type boolean and case label value of type boolean.
    e. switch expression of type int and case label value of type char.

8. Which one of the following for statements is valid? Select the one correct answer.

```
a.  for ( int i=10; i=0; i-- ) {}
b.  for ( int i=0, j=100; i<j; i++, j--) {;}
c.  int i, j; for ( j=100; i<j; j-- ) { i += 2; }
d.  int i=100; for ( (i>0); i-- ) {}
e.  int j=10; for ( int i=0, j+=90; i<j; i++ ) { j--; }
```

9. Which statements are valid when occurring on their own? Select the **three** correct answers.

```
a.  do { break; } while ( true );
b.  if ( true ) { break; }
c.  switch (1) { default: break; }
d.  for (;true;) break;
e.  while () break;
```