Integrating Sun Certification Objectives into an IS Programming Course

Akram Al-Rawi, Azzedine Lansari, Faouzi Bouslama
Zayed University, Abu Dhabi, UAE
akram.alrawi@zu.ac.ae, azzedine.lansari@zu.ac.ae, faouzi.bouslama@zu.ac.ae

Abstract

Information Systems Colleges are facing numerous challenges to develop and keep their curriculum current. Including certification objectives in IS courses give IS colleges the edge needed to remain competitive. Java, a modern object oriented programming language, has become a critical component of IS curricula because of its platform independence. This paper shows how to integrate objectives of the Sun Certified Programmer for Java 2 certification into a programming course. Current Java textbooks emphasize problem-solving concepts and do not cover certification objectives. After careful review of numerous textbooks, a textbook is selected to maximize coverage of the course and certification objectives. A detailed master syllabus is developed to optimize coverage of the certification objectives while maintaining consistency of the course. Upon completion of the course, students have the knowledge to take the certification exam and can demonstrate to employers that they have mastered the necessary skills to be productive.

Keywords: Sun certification, Java programming, Curriculum design, Information systems.

Introduction

The field of Information Systems (IS) is becoming increasingly competitive which is putting pressure on academic institutions to produce graduates that are productive as soon as they join the workforce. Currently, a number of employers are looking for IS graduates to acquire certification in addition to their Bachelor degree. Information Technology (IT) certifications give an edge to graduates as they demonstrate that they have acquired the necessary skills to be highly productive. IS colleges use the IS 2002 recommendation (Gorgone, Davis, Valacich, Topi, Feinstein, & Longenecker, 2002) to develop curricula that make them more competitive. The IS 2002 model curricula includes at least one course in a modern object oriented programming language. Adding certification objectives to the programming courses provides IS colleges with the potential to be more competitive.

ABET ("Criteria for Accrediting Computing Programs", 2004) is the agency in North America with the responsibility for accrediting all programs in computing. ABET has achieved worldwide recognition as an accreditation body and is increasingly called to review and accredit computing programs worldwide. The Computing Accreditation Commission (CAC) has responsibility for accrediting information systems programs. The IS 2002 model curriculum, which implements the ABET requirements for accreditation, states that students must be proficient in a modern object oriented language. Many institutions have adopted Java; a modern object oriented programming language, as a tool to teach problem solving and programming

concepts. Most IS programs offer at least two courses in Java.

With today's dynamic economy, industry and government are increasingly looking for certified professionals. There are over four hundred certifications in the IT field and the number is growing. The Sun certified programmer for Java 2 is one of these certifications. Community colleges have customized their curriculum toward certifications such as CompTIA A+, CompTIA Network+, or Cisco CCNA (Zeng, 2004). However, integrating certification into the Bachelor degree is still in its early stages.

Although there are at least 1200 textbooks for teaching Java as listed in www.amazon.com, the authors could not find any textbook that addresses the integration of the Sun Certification objectives. On the other hand, the authors found a few reference books that only address the certifications objectives (Mughal & Rasmussen, R., 2000). However, the certification objectives deal mainly with Java syntax. While textbook concentrate on the problem solving issues and ignore most of the Sun certification objectives. The Sun certification objectives by themselves may not fulfill the IS 2002 requirement to be proficient in an object oriented language. To be proficient in a language such as Java, students need a lot of practice in problem solving using that particular language. However, a number of companies are looking for professionals with certain certifications such as Sun certified Java programmer, Cisco CCNA, or CompTIA Network+ because they represent a standard measurements for certain skills.

This paper shows how to integrate objectives of the Sun Certified Programmer for Java 2 certification into a programming course. Current Java textbooks emphasize problem-solving concepts and do not cover certification objectives. After careful review of numerous textbooks, a textbook is selected to maximize coverage of the course and certification objectives. A detailed master syllabus is developed to optimize coverage of the certification objectives while maintaining consistency of the course. Upon completion of the course, students have the knowledge to take the certification exam and can demonstrate to employers that they have mastered the necessary skills to be productive. The rest of the paper is organized as follows. Section 2 introduces the Sun certification for Java programmer. Section 3 shows the mapping between the course topics and the certification objectives. Section 4 introduces the master syllabus for the complete course. Section 5 is the conclusion.

Sun Certified Programmer for Java 2

IS students may choose one or more certificate to prove to their employers that they have sufficient training experience to work in a field where specific skills are required. Benefits for becoming certified include:

- Exposure to the latest software, equipment or other knowledge-based advantages, individuals might not otherwise have.
- Increased level of expertise.
- Additional knowledge and skills that allow individuals to move into a new area or perform their current job more efficiently.
- Customer confidence based on individual evidence of qualifications and suitability for the task at hand or project put out for bids.
- Contact and networking with top-performing professionals in the field, around the world.

There are over four hundred certificates in the IT field. Many community colleges offer a variety of training courses to prepare students for certifications such as CompTIA A+, compTIA network+, and Cisco CCNA. Many four-year colleges have also started their own certification training programs to meet the demands of the local market. Almost all IS four years programs require

two to three courses in programming. Though, the programming language may vary from one institution to the other, Java is widely used because it is platform independent. Sun developed two certificates in Java. The first certificate is the Sun Certified Programmer for Java 2. The most recent version of the certification exam for the Sun Certified Programmer for Java 2 is Platform 1.4. It is a two-hour exam and consists of 61 multiple-choice questions. Currently, the passing score is 52%. The exam objectives, as published by Sun, consist of nine sections as shown in Appendix I. These objectives include the coverage of the following topics:

- Declarations and Access Control
- Flow control, Assertions, and Exception Handling
- Garbage Collection
- Language Fundamentals
- Operators and Assignments
- Overloading, Overriding, Runtime Type and Object Orientation
- Threads
- Fundamental Classes in the java.lang Package
- The Collections Framework

These objectives can be integrated into the course topics so that by the end of the semester students can fulfill the requirements of the course as well as taking the certification exam.

Mapping Certification Objectives into Course Topics

Most Java textbooks state that the main goal of the textbook is to introduce concepts in problem solving rather than the syntax of the language. Course instructors select their textbooks to fulfill the objectives of the course while focusing on teaching problem solving. However, if we wish to integrate the certification objectives into the course, selecting the textbook becomes very critical. Very few textbooks can be used to cover all certification objectives. A popular textbook such as "*Java How to Program*" (Deitel & Deitel, 2003) can be used to cover most of the certification objectives. On the other hand, the book titled "*Problem Solving with Java*" (Koffman, & Wolz, 2002) only covers about 65% of the certification objectives. In fact, comprehensive Java textbooks such as "*Java How to Program*" were written to be covered in two semesters. Most Java textbooks, however, are designed to be covered in one semester. These books even though they do not cover all certification objectives are, for a number of reasons, very well suited to teach the first programming course in the IS curriculum.

In this paper, we have selected the one-semester textbook titled "*Problem Solving with Java*", by Koffman and Wolz because of its early introduction of object-oriented programming concepts. Table 1 lists the contents of the textbook along with the certification objectives addressed by the textbook. Column two lists the certification objective numbers, which are described in Appendix I.

Table 1: Mapping certification objectives

Course Topics	Certification Objectives
Ch.1: Introduction to Computers, Problem Solving, and Programming Overview of Computers Computer Components Computer Software	

Processing a High-Level Language Program The Software Development Method	
Ch.2: Using Primitive Data Types and Using Classes Primitive Data Types Processing Numeric Data Introduction to Methods The *String* class Input/Output with class *JOptionPane* and method *println()* Anatomy of a program Numerical computations with class *Math*	8.1 8.2
Ch.3 Object-Oriented Design Manipulating String Objects Processing Integers Review of Methods Using Multiple Classes Formatting output *Applets*, AWT, and the *Graphics* class	1.2 1.3 1.4 4.1 4.4 4.5 4.6, 5.4
Ch.4: Control Structures: Decisions and loops Control Structures *Boolean* Expressions The *if* statement Decision Steps in Algorithms Nested *if* and *switch* *while* and *for* statements State-Controlled Loops	2.1 2.2 5.1 5.3
Ch.5 Arrays and Vectors Array declarations Processing Arrays and Array Elements Operations on Whole Arrays Searching and Sorting an Array Arrays of Objects Multidimensional Arrays Vectors Wrapper Classes for Primitive Type Data The *Arrays* and *ArrayList* Collection Classes	1.1 1.4 4.3 5.2 8.3
Ch.6: Class Hierarchies, Inheritance, and Interfaces Class Hierarchies and Inheritance Operations in a Class Hierarchy Polymorphism Interfaces Abstract Classes Drawing Figures Using an Abstract Class and an Interface Packages and Visibility	1.2 4.1 4.2 6.1 6.2 6.3
Ch.7 Graphical User Interfaces AWT, *Swing*, and Browser-Applet Interaction The Java Event Model Using a GUI in an Application Components for Making Choices Listener Classes as Inner Classes Layout Managers	
Ch.8: Exceptions, Streams, and Files Exceptions Stream and Text Files Using Text Files Binary Files	2.3 2.4

Ch.9 Recursion Recursive Methods Recursive Mathematical Methods Use of the Stack in Recursion Recursive Methods with Arrays, Vectors, and Strings Binary Search	
Ch.10: Linked Data Structures Linked Lists Stacks Queues Binary Trees A Binary Search Tree Class *LinkedList* Collection Class and *List* Iterators	9.1 9.2

Master Course Syllabus with Certification Objectives

Examination of how the certification objectives map into the textbook topics as listed in Table 1 shows that some objectives are not covered by the selected textbook. Objectives 2.5 and 2.6 are addressed only by textbooks that address Java 1.4. Objectives 3.1, 3.2, 3.3, 7.1, 7.2, 7.3, and 7.4 are not addressed by the selected textbook. Some chapters in the textbook do not contribute to the certification objectives such as Chapter 9. Some of the Certification objectives, such as multi-threading, may not be covered by the first programming course in Java in the IS curriculum. On the other hand, recursion is normally taught in the second programming course. One approach to include all the certification objectives is to drop the recursion chapter and add a chapter on multi-threading (objectives 7.1-7.4). Garbage collection objectives (3.1-3.3) can be easily addressed in Chapter 3. Ideally, it will be better to adopt a textbook that addresses the certification objectives as well as the treatment of problem solving and early coverage of the object orientation. However, at the present time such a textbook is not available in the market. A master course syllabus that covers the course topics and integrates the certification objectives is shown in Table 2.

Table 2: Master Course Syllabus

Problem Solving & Programming Using Java

Credit hours: 4

Course Description:
This course introduces students to basic concepts in program design and development using Java. A disciplined approach to problem solving and algorithm development using object-oriented concepts is stressed. Topics include syntax and semantics, input/output, selection, iterative constructs, methods, classes, data types, arrays, strings, multithreading, exceptions, file processing, garbage collections, graphical user interface, the collection classes. The course also covers objectives of the Sun Certified Programmer for Java 2 allowing students to take the certification exam upon completion.

Textbook: Problem Solving with Java, 2nd edition
Author: Koffman & Wolz
Publisher: Addison-Wesley, 2002

Reference: Java How To Program, 5th Edition
Author: Deitel & Deitel
Publisher: Prentice Hall, 2003

Course Objectives:
To understand the concepts of problem solving
To understand the five-step approach to program design
To be able to successfully utilize the subset of the Java programming language

To write programs that adhere to practices of software development
To be able to independently design algorithms to solve real-world problems
To have a basic understanding of principles of the Object Oriented design
To complete the Sun certification objectives.
To take the Sun Certification Programmer for Java 2 exam

Evaluation Procedures

Three Tests (closed book)	40%
Homework/Projects	20%
Certification Exam	40%

Course Topics and Certification Objectives

Week/ Period	Course Topics	Book/ Sections	Certif. Object. (App. I)
1.1	Int. to Computers, problem solving	1.1, 1.2 1.3	
1.2	Processing a High-Level language program	1.4, 1.5	
1.3	Using Primitive Data Types and Using Classes	2.1, 2.2	
1.4	Introduction to Methods, The String class	2.3, 2.4	
2.1	Input/Output with class JOptionPane and method println()	2.5	8.1
2.2	Problem Solving in Action, Anatomy of a Program	2.6, 2.7	8.2
2.3	Numerical computation with class Math	2.8	
2.4	Common Errors and Debugging	2.9	
3.1	Object-Oriented Design and Writing Worker Classes	3.1	1.2, 1.3
3.2	A Worker Class That Manipulates String Objects	3.2, 3.3	1.4, 4.1
3.3	Review of Methods, Using Multiple Classes	3.4, 3.5	4.4, 4.5
3.4	Formatting Output, Applets and Graphics Class	3.6, 3.7	4.6, 5.4
4.1	The if Statement	4.3	2.1
4.2	Decision Steps in Algorithms	4.4	2.2
4.3	Multiple-Alternative Decisions: Nested if and Switch	4.5	5.1, 5.3
4.4	Test # 1 (closed book)		
5.1	Counting Loops, while and for Statements	4.6	1.1, 1.4
5.2	State Controlled Loops	4.7	4.3 ,5.2
5.3	Putting It All Together: Case Study	4.8	8.3
5,4	Debugging and Testing Programs With Decisions and Loops	4.9, 4.10	
6.1	Arrays and Vectors	5.1	1.2, 4.1
6.2	Processing Arrays and Array Elements	5.2	4.2 , 6.1
6.3	Operations on Whole Arrays	5.3	6.2, 6.3
6.4	Searching and Sorting an Array	5.4	
7.1	Arrays of Objects	5.5	
7.2	Multidimensional Arrays	5.6	
7.3	Vectors	5.7	
7.4	Wrapper Classes for Primitive Type Data	5.8	
8.1	The Arrays and ArrayList Collection Classes`	5.9	
8.2	Class Hierarchies, Inheritance, and Interfaces	6.1	1.2 , 4.1
8.3	Operations in a class Hierarchy	6.2	4.2, 6.1
8.4	Polymorphism, Interfaces	6.3, 6.4	6.2, 6.3
9.1	Test # 2 (closed book)		
9.2	Abstract Classes	6.5	
9.3	Drawing Figures Using an Abstract class and an Interface	6.6	
9.4	Review of Certification Objectives		
10.1	Packages and Visibility, Testing a Program System	6.7, 6.8	
10.2	Graphical User Interfaces	7.1	
10.3	Designing a first GUI	7.2	
10.4	The Java Event Model, Using a GUI in an Application	7.3, 7.4	
11.1	Components for Making Choices	7.5	
11.2	Designing a GUI for an Existing class: Case Study	7.6	

11.3 11.4	No Class No Class		
12.1 12.2 12.3 12.4	Listener Classes as Inner Classes, Layout Managers Exceptions, Streams, and Files Streams and Text Files, Using Text Files Review of Test 3	7.7, 7.8 8.1 8.2, 8.3	 2.3 2.4
13.1 13.2 13.3 13.4	A GUI for Processing the DVD Collection, Binary Files Test # 3 (closed book) Introduction to Multi Threading (Deitel & Deitel) Thread Priorities and Thread Scheduling	8.4, 8.5 15.1, 15.2 15.3, 15.4	 7.1, 7.2 7.3, 7.4
14.1 14.2 14.3 14.4	Thread Synchronization Daemon Threads, Runnable Interface Finalizers, Static Class Members (Deitel & Deitel) Collections Overview (Deitel & Deitel)	15.5, 15.6 15.9, 5.10 8.14, 8.15 21.2, 21.3	 3.1,3.2 3.3 9.1, 9.2
15.1 15.2 15.3 15.4	Interface Collection and Class Collections Lists Sets, Maps Review of the certification exam	21.4 21.5 21.7, 21.8	
16.1	Certification Exam (closed book)		

The above master course syllabus is designed to achieve the course objectives as well as prepare students to take the Sun Certified Programmer for Java 2 certification exam. The proposed master syllabus is designed for a four-credit hour course to allow instructors the flexibility to include detailed coverage of the certification objectives when needed. The three exams are designed to verify comprehension of problem solving and object oriented programming concepts. The course assessment is designed to cover both comprehension of the course material and the certification objectives. The assessment is optimized by assigning 60% of the final grade to the course work and 40% to the certification exams. Obtaining a passing grade in the course is necessary but not a sufficient condition to pass the certification exam.

During fall 2004 semester, the author taught a programming and problem-solving course using Java. In the syllabus, the course objectives included taking the Sun Certified Programmer for Java 2 exam as an important part of the course evaluation. Upon completion of the course, the author and the students took the Sun Certified Programmer for Java 2 exam. Even though the certification exam was quite complex and the time allowed for answering questions was very short (one minute per multiple choice/multiple answer question), the instructor and two third of the class passed the exam. Currently, the author is teaching a similar class and anticipates a higher passing rate as he is aware of the difficulties encountered while taking the exam. Similarly, the author is teaching the CCNA certification objectives in a network class with the intent to take the certification exam at the end of the spring 2005 semester. In order to be able to take the CCNA exam, which requires the use of Cisco hardware and software, a lab was equipped with Cisco equipment. The author anticipates a high passing rate at the completion of the course.

Conclusion

This paper presents an approach to combine course and certification objectives to provide an avenue for students to acquire certification upon completion of the course. There are about 1200 Java textbooks in the market. But none of them address explicitly the certification objectives. There are a few reference books that are written to basically review the certification objectives. However, these books are not useful as textbooks and cannot be adopted for a programming course. The aim of this study is to generate a course syllabus that primarily covers the course objectives and also includes the certification objectives when needed. Textbook selection is based primarily

on how problem solving is covered and on the early introduction of the object orientated programming concepts. The Sun certification objectives are reviewed and mapped into the table of content of each textbook. However, no textbook completely covers the certification objectives. After careful consideration of all the factors, the selected textbook covers the object oriented programming concepts at an early stage and allows coverage of most certification objectives A course syllabus is developed to integrate the certification objectives into the textbook topics. This syllabus is relatively extensive and the amount of time to cover the whole syllabus may take more than one semester. Covering the syllabus in two semesters may not be feasible because some key concepts have to be repeated before taking the certification exam. Therefore, a modified syllabus is proposed which excludes course topics such as recursion, which is not critical for the certification. To test this proposal, the author and a group of students took the Sun certification exam upon completion of the Java programming course. The author and two thirds of the class passed the certification exam. Currently the author is testing the possibility of integrating the CCNA objectives into a networking class. Providing students with an avenue to take certification in computing, makes IS colleges more attractive to students as IT certifications have become valued assets for individuals seeking employment or advancement in the computing field.

References

Criteria for Accrediting Computing Programs, Computing Accreditation Commission. Retrieved February 15, 2005 from http://www.abet.org/criteria_cac.html and http://www.abet.org/images/Criteria/C001%2004-05%20CAC%20Criteria%2011-18-03.pdf

Deitel, H., & Deitel, P. (2003). *Java How to program* (5th ed.). Prentice-Hall.

Gorgone, J., Davis, G., Valacich, J., Topi, H., Feinstein, D. & Longenecker, H. (2002). *IS 2002: Model curriculum and guidelines for undergraduate programs in information systems.* Retrieved February 15, 2005 from http://www.acm.org/education/is2002.pdf

Koffman, E. & Wolz, U. (2002). *Problem solving with Java* (2nd ed.). Addison-Wesley.

Mughal, K. & Rasmussen, R. (2000). *A programmer's guide to Java certification*, Addison-Wesley.

Zeng, F. (2004). A new approach to integrate computer technology certification into computer information system programs. *Annual ASEE Conference*, Salt Lake City, Utah, Session 2558. Retrieved February 15, 2005 from http://www.asee.org/acPapers/2004-1708_Final.pdf

Appendix I

Sun Certified Programmer for Java 2 Platform 1.4 Exam Objectives

Section 1: Declarations and Access Control

1.1 Write code that declares, constructs and initializes arrays of any base type using any of the permitted forms both for declaration and for initialization.

1.2 Declare classes, nested classes, methods, instance variables, static variables and automatic (method local) variables making appropriate use of all permitted modifiers (such as public, final, static, abstract, etc.). State the significance of each of these modifiers both singly and in combination and state the effect of package relationships on declared items qualified by these modifiers.

1.3 For a given class, determine if a default constructor will be created and if so state the prototype of that constructor.

1.4 Identify legal return types for any method given the declarations of all related methods in this or parent classes.

Section 2: Flow control, Assertions, and Exception Handling

2.1 Write code using if and switch statements and identify legal argument types for these statements.

2.2 Write code using all forms of loops including labeled and unlabeled, use of break and continue, and state the values taken by loop counter variables during and after loop execution.

2.3 Write code that makes proper use of exceptions and exception handling clauses (try, catch, finally) and declares methods and overriding methods that throw exceptions.

2.4 Recognize the effect of an exception arising at a specified point in a code fragment. Note: The exception may be a runtime exception, a checked exception, or an error (the code may include try, catch, or finally clauses in any legitimate combination).

2.5 Write code that makes proper use of assertions, and distinguish appropriate from inappropriate uses of assertions.

2.6 Identify correct statements about the assertion mechanism

Section 3: Garbage Collection

3.1 State the behavior that is guaranteed by the garbage collection system.

3.2 Write code that explicitly makes objects eligible for garbage collection.

3.3 Recognize the point in a piece of source code at which an object becomes eligible for garbage collection.

Section 4: Language Fundamentals

4.1 Identify correctly constructed package declarations, import statements, class declarations (of all forms including inner classes) interface declarations, method declarations (including the main method that is used to start execution of a class), variable declarations, and identifiers.

4.2 Identify classes that correctly implement an interface where that interface is either java.lang.Runnable or a fully specified interface in the question.

4.3 State the correspondence between index values in the argument array passed to a main method and command line arguments.

4.4 Identify all Java programming language keywords. Note: There will not be any questions regarding esoteric distinctions between keywords and manifest constants.

4.5 State the effect of using a variable or array element of any kind when no explicit assignment has been made to it.

4.6 State the range of all primitive formats, data types and declare literal values for String and all primitive types using all permitted formats bases and representations.

Section 5: Operators and Assignments

5.1 Determine the result of applying any operator (including assignment operators and instance of) to operands of any type class scope or accessibility or any combination of these.

5.2 Determine the result of applying the Boolean equals (Object) method to objects of any combination of the classes java.lang.String, java.lang.Boolean and java.lang.Object.

5.3 In an expression involving the operators &, |, &&, || and variables of known values state which operands are evaluated and the value of the expression.

5.4 Determine the effect upon objects and primitive values of passing variables into methods and performing assignments or other modifying operations in that method.

Section 6: Overloading, Overriding, Runtime Type and Object Orientation

6.1 State the benefits of encapsulation in object oriented design and write code that implements tightly encapsulated classes and the relationships "is a" and "has a".

6.2 Write code to invoke overridden or overloaded methods and parental or overloaded constructors; and describe the effect of invoking these methods.

6.3 Write code to construct instances of any concrete class including normal top level classes and nested classes

Section 7: Threads

7.1 Write code to define, instantiate and start new threads using both java.lang.Thread and java.lang.Runnable.

7.2 Recognize conditions that might prevent a thread from executing.

7.3 Write code using synchronized wait, notify and notifyAll to protect against concurrent access problems and to communicate between threads.

7.4 Define the interaction among threads and object locks when executing synchronized wait, notify or notifyAll.

Section 8: Fundamental Classes in the java.lang Package

8.1 Write code using the following methods of the java.lang.Math class: abs, ceil, floor, max, min, random, round, sin, cos, tan, sqrt.

8.2 Describe the significance of the immutability of String objects.

8.3 Describe the significance of wrapper classes, including making appropriate selections in the wrapper classes to suit specified behavior requirements, stating the result of executing a fragment of code that includes an instance of one of the wrapper classes, and writing code using the following methods of the wrapper classes (e.g., Integer, Double, etc.):

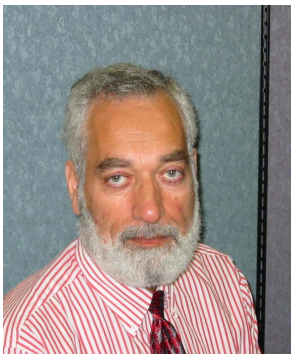
doubleValue, floatValue, intValue, longValue, parseXxx , getXxx , toString , toHexString

Section 9: The Collections Framework

9.1 Make appropriate selection of collection classes/interfaces to suit specified behavior requirements.

9.2 Distinguish between correct and incorrect implementations of hashcode methods.

Biographies



Akram Al-Rawi is a Professor of IS at Zayed University, UAE. He has worked at several academic institutions of which the last two were the University of Missouri-Columbia and Columbia College, MO. His teaching interests include programming languages, logic design, and computer architecture. His research interests include computer simulation, web-caching architecture, and curriculum design. He holds certificates in ICDL, CompTIA A+, Sun Certified Programmer for Java 2, and CCNA1.

Azzedine Lansari received a PhD in Biomedical Engineering from North Carolina State University in 1992. From 1992-1998, he was a senior researcher at MANTECH, NC. He joined Zayed University in August 1998. His research interests include systems modeling, educational technology and curriculum design in Information Systems. His teaching interests include Instructional Technology and statistical modeling.

Faouzi Bouslama is a Professor of IS at Zayed University. He received a PhD in Electronics Engineering from Shizuoka University in 1992. From 1992-1994, he was a researcher at Toshiba Co., Tokyo. From 1994-2000, he was Associate Professor of Information Systems, Hiroshima City University, Japan. He joined Zayed University, UAE, in August 2000. His research interests include Neural Networks, Fuzzy Logic, Curriculum Design, and System Modeling.