

Integrating Certifications into Information Systems Curricula

Akram Al-Rawi, Azzedine Lansari, and Faouzi Bouslama
College of Information Systems, Zayed University
P. O. Box 4783, Abu Dhabi, UAE

Abstract

The objective of this paper is to propose a curriculum that provides graduates with an IS degree as well as the opportunity to acquire IT certificates upon graduation. In this paper, we examine institutions which integrate certificates in their IS curriculum. Moreover, we examine the IS 2002 report and ABET requirements to identify IS tracks that can lead to a particular certification. Popular IT certificates are identified and for each certificate a series of IS courses are developed to include the objectives of that particular certificate. For example, we will show how to integrate the Sun Certification for Java Programmer into programming courses, the Cisco CCNA or Network+ certificate into networking course, the A+ certificate into the IT Hardware and Software course and an operating system course, and the MOUS certificate into the personal productivity with IS technology course. Other certifications such as ORACLE can also be included into the IS elective courses. The goal is for the curriculum to provide an avenue for students to take the certification exam after completing the course series leading to that certificate. This research will help faculty design their courses to integrate certain certificates. It will also help students choose their courses in order to acquire certificates.

1. Introduction

Information Systems (IS) programs are accredited by oversight bodies that determine if the program provides adequate education. This assures that, as long as professionals graduate from accredited programs, they will start their professional lives with the knowledge they need to perform effectively. Currently, ABET¹ is the accreditation body with responsibility for accrediting all programs in computing. The Computing Accreditation Commission (CAC) has responsibility for accrediting IS programs. The IS 2002² model curriculum provides recommendation for the core IS curriculum to meet ABET criteria. While ABET specifies the 120 credit hours for the Bachelor degree, IS 2002 addresses 33 credit hours only.

Growing demand for Information Technology (IT) skills from industry and government is putting pressure on academic institutions to produce graduates that have the necessary skills to be productive as they join the workforce. Many employers are looking for IS graduates to acquire certification in addition to their Bachelor degree. IS graduates can demonstrate that they have the necessary skills to be productive by acquiring one or more IT certificates. The IEEE and ACM recognize the importance of the IT certification and provide their members with over four hundred online courses leading to technical certifications.

In this paper, a comprehensive model curriculum that fulfills the IS 2002 recommendations, the ABET criteria and the integration of certifications is proposed. In order to meet the accreditation

requirements, the proposed model curriculum uses the IS 2002 recommendations, which states that a minimum of 30 credit hours in IS are required. Moreover, the model addresses all ABET criteria, which provides recommendations for all the components including general education (GE), major core courses, major electives and general elective courses. The proposed model is based on the above requirements and provides a framework to integrate certification into IS courses at various levels of the model curriculum.

The rest of the paper is organized as follows: Section 2 introduces the ABET requirements, section 3 introduce the IS 2002 module curricula, section 4 provides the benefits of IT certifications, section 5 discusses the issue of integrating the certification objectives into the IS curriculum, section 6 is a case study, and section 7 is the conclusion.

2. ABET Requirement for Accreditation

ABET is an organization that accredits programs in engineering and technology. ABET has become the main program accreditation body in the US and currently in a number of other countries. ABET provide general requirement for the IS curriculum, these are:

- The curriculum must include at least 30 semester-hours of study in information systems topics.
- The curriculum must contain at least 15 semester-hours of study in information systems environment, such as business.
- The curriculum must include at least 9 semester-hours of study in quantitative analysis.
- The curriculum must include at least 30 semester-hours of study in general education to broaden the background of the student.

Detailed description of each general requirements are summarized below:

Information Systems:

- All students must take a broad-based core of fundamental information systems material consisting of at least 12 semester hours.
- The core materials must provide basic coverage of the hardware and software, a modern programming language, data management, networking and telecommunications, analysis and design, and the role of IS in organizations.
- Theoretical foundations, analysis, and design must be stressed throughout the program.
- Students must be exposed to a variety of information and computing systems and must become proficient in one modern programming language.
- All students must take at least 12 semester hours of advanced course work in information systems that provides breadth and builds on the IS core to provide depth.

Information Systems Environment:

- The 15 semester hours must be a cohesive body of knowledge to prepare the students to function effectively as an IS professional in the IS environment such as business.

Quantitative Analysis:

- The curriculum must include at least 9 semester-hours of quantitative analysis beyond pre-calculus.
- Statistics must be included.
- Calculus or discrete mathematics must be included.

Additional Areas of Study:

- The oral and written communications skills of the student must be developed and applied in the program.
- There must be sufficient coverage of global, economic, social and ethical implications of computing to give students an understanding of a broad range of issues in these areas.
- Collaborative skills must be developed and applied in the program.

3. IS 2002 Model Curricula

The IS 2002 Model Curricula implement the core of the IS curriculum, so that the curriculum can be accredited by ABET. They provide a detailed description of 30 semester hours plus a 3 semester hours of prerequisite. These 33 semester hours are listed below:

- 0- Personal Productivity with IS Technology
- 1- Fundamentals of Information Systems
- 2- Electronic Business Strategy, Architecture and Design
- 3- Information Systems Theory and Practice
- 4- Information Technology Hardware and System Software
- 5- Programming, Data, File and Object Structures
- 6- Networks and Telecommunication
- 7- Analysis and Logical Design
- 8- Physical Design and Implementation with DBMS
- 9- Physical Design and Implementation in Emerging Environments
- 10- Project Management and Practice

A typical Bachelor degree in IS includes:

- 39 hours of general education
- 9 hours of quantitative analysis
- 33 hours of IS core
- 15 hours of IS environment
- 15 hours of IS electives
- 15 hours of general electives

These will be added to make the 120-126 hours, and the implementation may vary.

4. IT Certification

IT certification is the process of mastering material pertaining to a particular hardware system, operating system, programming language, or other software program, then proving your mastery

by passing a series of exams. Certification programs are developed and administered either by a manufacturer or a professional organization such as the Computing Technology Industry Association (CompTIA). Student can pursue a number of different certifications, depending on student specialty interest. For example to prove a mastery of many aspects of networking, student can choose to become Network+ certified.

Benefits for becoming certified may come in the form of:

- Additional knowledge and skills that allow individuals to move into a new area or perform current job more effectively.
- Exposure to the latest software, equipment or other knowledge.
- Increased level of expertise
- Contact and networking with top-performing professionals in the field, around the world.
- Customer confidence based on the evidence of qualifications.

The IEEE and ACM recognized the importance of certification and provide its members with over four hundred online courses leading to professional certifications, such as CompTIA Network+ and Sun Certified Programmer for Java 2. Each certificate has certain objectives. There are vendors such as Sun Microsystems and Microsoft which offer training courses for their certification. There are also training institutions which offer courses for almost any certificates. There are also community colleges that design their program to concentrate on a specific field, such as network administration, PC technician, Help Desk technician and Customer Service Technician³. Quite few universities also start offering training which leads to certifications³. It is the objectives of this paper to address the issue of integrating the certification objectives into the IS curriculum so that the student can earn a bachelor degree in IS and one or more IT certificate.

5. Integrating IT Certification into IS Courses

All IT certifications have specific objectives and require mastering certain skills that are not necessarily included in IS curricula. On the other hand, IS courses have specific objectives and outcomes that need to be achieved in order to pass the course and fulfill the accreditation requirement. In this study four areas in the IS curriculum which can be strengthened by IT certification have been identified: personal productivity tools, networking, programming and problem solving, and hardware and software. Furthermore, five potential IT certificates that can be integrated in these areas have been identified, as shown in Table 1. These are MS MOUS, CompTIA Network+, Cisco CCNA, CompTIA A+, and Sun Certified Programmer for Java 2 platform.

The Microsoft Office User Specialist (MOUS) is an IT certificate that measures and validates skills in the MS Office suite. The MOUS certificate objectives can be easily integrated into the personal productivity with IS Technology course (IS 2002.P0), which is a prerequisite course in the IS 2002 model curriculum². IS 2002.P0 emphasizes personal productivity concepts using functions and features in computer software such as spreadsheets, databases, presentation graphics and web authoring. This course requires prior elementary knowledge of word processing, spreadsheet, email, and web browsing. The IS 2002.P0 objectives can integrate the objectives of the MOUS certificate which include the mastery of Microsoft Word, Excel, Access, PowerPoint, and Outlook. The MOUS offers various levels of expertise including: Master,

Expert and Specialist. The master certification requires the candidates to successfully complete three required exams (Word 2002 Expert, Excel 2002 Expert, PowerPoint 2002) and one elective exam (either Access 2002 or Outlook 2002).

The computing technology industry association (CompTIA) provides six certificates including the A+ and the Network+. The A+ is designed to provide a competency level for personal computer repair and maintenance technicians. The A+ certification insures that the IT professional has obtained the necessary skills to configure, maintain, troubleshoot personal computers, as well as install operating systems and peripheral devices such as printers. The A+ certification objectives can be integrated into the IT Hardware and System Software course (IS 2002.4) as well as the Operating System (OS) course which is an IS elective. The IS 2002.4² provides students with an exposure to IT hardware and software components and their interaction. Systems components and peripheral devices are identified and studied. The OS software, including I/O drivers and extensions to the OS is studied and utilized in the laboratory. The OS course can be used to integrate and emphasize the OS part of the certification objectives. The A+ has two separate exams the core exam, which can be taken in the hardware and software course and the OS exam, which can be taken upon completion of the OS course. Therefore, in order to take the A+ certification students need to enroll in the OS course in addition to the IT hardware and software course.

Network+ is a professional certification established by CompTIA that verifies broad, vendor-independent networking technology skills such as an understanding of protocols, topologies, networking hardware, and network troubleshooting^{4,5}. Network+ may also be a stepping stone to more advanced certifications. For example, Novell now accept Network+ as a substitute for its Networking Technologies exam for candidates pursuing CNE status. Certified Network+ professionals know the layers of the OSI model, can describe features and functions of network components, and have the skills needed to install, configure and troubleshoot basic networking hardware peripherals and protocols. The A+ certification is recommended before taking the Network+ certification. The IS 2002.6 (Networks and Telecommunication) course provides an in-depth knowledge of data communications and networking requirements including network and telecommunication technologies, hardware and software. The Network+ certification domain areas are: media and topologies, protocols and standards, network implementation, and network support. The Network+ certification domain areas that can be easily integrated into the IS 2002.6 course are: media and topologies, protocols and standards, and network implementation. The last domain area, network support, can be addressed in an elective network course.

An alternative to the Network+ certification is the Cisco CCNA certification^{6,7}. The objectives of the Cisco CCNA certification can be covered in two Network courses. The Cisco CCNA topics are OSI reference model and layered communications, network protocols, routing, WAN protocols, network management, LAN design, Cisco networking basics, and bridging and switching. Even though the Cisco Academy teaches the CCNA curriculum in four semesters, the certification objectives can be covered in two courses, including a lab component. The following topics can be integrated into the IS2002.6: The OSI model, network devices, TCP/IP and IP addressing, network topology and design, WAN concepts, routers and IOS basics, and router start up and configuration. The second network course includes the following topics: routing protocols, IPX/SPX, access lists, PPP and ISDN, frame relay, and switching and VLANs.

While Network+ is a non proprietary certification, Cisco CCNA requires the use of Cisco products such as routers and switches. Therefore the lab for this course should be equipped with Cisco devices.

Sun developed the Certified Programmer for Java 2 certificate to ensure that the programmer masters specific skills to be productive. The certification topics are: declarations and access control; flow control, assertions and exception handling; garbage collection; language fundamentals; operator assignment; overloading, overriding, runtime type and object orientation; threads; fundamental classes in the java language package; and the collections framework⁸. The integration of these topics into the IS 2002.5 course is discussed in the case study section.

Table 1: IT Certification for Integration into IS Curriculum

IT Certificate	Certification Exams	Matching IS Courses
MS MOUS	Master, Expert or Specialist exams	IS 2002.P0
CompTIA A+	Core exam, OS exam	IS2002.4 and one elective Operating Systems course
CompTIA Network+	One 2-hour exam (A+ is a recommended prerequisite)	IS 2002.6 (Networks and Telecommunication) and one elective network course
Cisco CCNA	One 2-hour exam	IS 2002.6 (Networks and Telecommunication) and one elective network course
Sun Certified Programmer for Java 2	One 2-hour exam (59 multiple choice questions)	IS 2002.5 (Programming and Object Structures)

6. Case Study

This case study shows how to integrate the Sun Certification for Java Programmer objectives, as developed by Sun and shown in Appendix I, into one programming course of the IS curriculum. This course corresponds to the IS2002.5 Programming, Data, File and Object Structures course. The IS2002.5 course topics are listed below:

Data structures and representation: characters, records and files; precision of data; information representation, organization, and storage; algorithm development; programming control structures; program correctness, verification, and validation; file structures and representations, programming in traditional and visual development environment that incorporate event-driven, object-oriented design.

Upon examination of the IS2002.5 course topics and the Sun certification objectives listed in Appendix I, a textbook⁹ is selected. There are over a thousand textbooks and reference books available for teaching programming with Java. Only a few textbooks can be used to cover most of the certification objectives. In this case study, the selected textbook covers about 70% of the certification objectives. After careful examination of the textbook topics and the Sun certification

objectives, Table 2 is generated to map the certification objective into the textbook topics. A master course syllabus that addresses both the textbook as well the certification objectives is then generated. Additionally, the certification objectives which are not covered by the textbook are listed. An examination of the chapter by chapters objectives along with the certification objectives shows that some of the certification objectives are not covered by the book such as 7.1, 7.2, 7.3, 7.4 (Threads), 2.5, 2.6, 9.1, and 9.2. Objectives 3.1, 3.2, and 3.3 are not covered in

Table 2: Textbook and Related Certification Objectives

Course Topics/Objectives	Cert. Objectives
Ch.1 Introduction to Computers and Java Overview of computer hardware and software Introduce the basic techniques of program design and object oriented programming Overview of the Java programming language	
Ch.2 Primitive Types, Strings, and Console I/O Become familiar with the Java primitive data types Learn the assignment statement and expressions Learn the Java data type used for strings, and do simple string processing Learn about simple keyboard input and screen output using System and Swing	4.1, 4.3 4.4, 4.6 5.1, 5.2 5.3, 8.2
Ch.3 Flow of Control Learn Java branching statements Learn loops Learn the type boolean	2.1, 2.2 5.1, 5.3
Ch.4 Defining Classes and Methods Learn the concepts of a class and of an object that instantiates the class Learn how to define classes in Java Learn to define and use methods in Java Learn to create objects in Java Learn how parameters work in Java Learn information hiding and encapsulation Learn the notion of reference, understand class variables and class parameters.	1.2, 1.4 1.5, 4.1 5.4
Ch.5 More About Objects and Methods Become familiar with more techniques for programming with classes and objects Learn static methods and static variables Learn overloading a method name Learn to define constructor methods in Java Learn packages and import statements in Java Learn to design methods using the top-down design technique Learn a number of techniques for testing methods including the use of driver	1.2, 1.3 1.4, 1.5 4.1, 4.3 4.4, 5.4 8.1, 8.3 5.2, 5.3
Ch.6 Arrays Find out what an array is and how to use arrays in simple Java programs Learn how to use array parameters and how to define methods that return an array	1.1, 1.5 4.4

<p>Learn the proper way to use an array as an instance variable in a class</p> <p>Introduce the topic of sorting an array</p> <p>Become familiar with multidimensional arrays</p>	
<p>Ch.7 Inheritance</p> <p>Become familiar with inheritance in general</p> <p>Learn how to define and use derived classes in Java</p> <p>Learn dynamic binding and polymorphism in general and in Java</p>	<p>6.1, 6.2 6.3</p>
<p>Ch.8 Exception Handling</p> <p>Become familiar with the notion of exception handling</p> <p>Learn Java syntax for exception handling</p> <p>Be able to use exception handling effectively in your own classes and programs</p>	<p>2.3, 2.4</p>
<p>Ch.9 Streams and File I/O</p> <p>Become familiar with the concept of an I/O stream</p> <p>Understand the difference between binary files and text files</p> <p>Learn how to save data in a file, using a Java program</p> <p>Learn how to read data from a file, using a Java program</p> <p>Learn how to use the classes ObjectOutputStream and ObjectInputStream to write and read, respectively, class objects with binary files</p>	
<p>Ch.10 Dynamic Data Structures</p> <p>Become familiar with vectors and how they are used in Java</p> <p>Learn what a linked data structure is and how it can be realized in Java</p> <p>Find out how to manipulate linked lists</p> <p>Learn to use inner classes in defining linked data structures</p> <p>Learn what iterators are and how to create and use them</p>	<p>3.1, 3.2 3.3, 4.1 9.1, 9.2 2.3, 2.4</p>
<p>Ch.11 Recursion</p> <p>Become familiar with the idea of recursion</p> <p>Learn to use recursion as a programming tool</p> <p>Become familiar with the binary search algorithm as an example of recursion</p> <p>Become familiar with the merge sort algorithm as an example of recursion</p>	
<p>Ch.12 Window Interfaces Using Swing</p> <p>Learn the basics of event-driven programming</p> <p>Learn to design and code a simple GUI that includes buttons and text</p> <p>Learn about the Swing classes BorderLayout, Color, Container, FlowLayout, GridLayout, JButton, JComponent, JFrame, JLabel, JPanel, JTextArea, JTextField, WindowAdapter, and others.</p>	<p>2.3, 2.4</p>
<p>Ch.13 Applets and HTML</p> <p>Find out how to write applets</p> <p>Learn to write simple HTML document</p> <p>Learn how to embed an applet in an HTML document</p>	<p>4.2</p>
<p>Ch.14 More Swing</p> <p>Learn to add menus, icons, borders, and scroll bars to your GUIs</p> <p>Understand the BorderLayout manager and Box class</p> <p>Understand some uses and advantages of inner classes</p> <p>Learn about the WindowListener interface</p> <p>Find out how to create GUIs with components that change from visible to invisible</p>	<p>1.2</p>

enough depth to satisfy the certification objectives. Another observation is that recursion is normally covered in the data structure course, and therefore can be removed from the course syllabus. Instead of covering recursion, instructors should cover multithreading to satisfy the certification objectives. An additional textbook¹⁰ is used to address those objectives and cover in more depth the certification objectives 3.1, 3.2, and 3.3.

The proposed master course syllabus shows the weekly coverage of the textbooks and the certification objectives. Moreover, the syllabus shows how assessment is distributed. The 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. 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. The certification exam consists of 59 multiple choice questions. The exam time is 120 min and the current passing score is 53%. Obtaining a passing grade in the course is necessary but not a sufficient condition to pass the certification exam.

7. Conclusion

In this paper, a new approach to develop IS curriculum which addresses the needs of the local industry of IT certifications while fulfilling the IS 2002 recommendations and ABET criteria was proposed. Five key IT certifications were identified for the purpose of integration into the IS 2002 core curriculum. The MOUS certification objectives were mapped into the IS 2002.P0 curriculum. The CompTIA A+ certification objectives were integrated into the IS 2002.4 and an IS operating systems course elective. The CompTIA Network+ was integrated into the IS2002.6 and one IS Networking course elective. Alternatively, the Cisco CCNA certification objectives may be mapped into similar IS courses. The Sun Certified Programmer for Java 2 certification objectives are mapped into one IS 2002.5 programming course. Selecting the appropriate textbook to teach the proposed IS curriculum is critical. Finding textbooks which fulfill the catalog description of the IS 2002 and the certifications objectives may be challenging as only a few textbooks meet these requirements. Particularly, it is easier to find the appropriate textbook to cover networking certifications than programming certifications. The case study showed how the Sun certified programmer for Java 2 objectives can be integrated into the IS 2002.5 programming course. Using these guidelines a master course syllabus can be generated to allow completion of the certification objectives in one semester. Students can take the certification exam upon completion of the course. By providing students with an avenue to take certification in computing, IS colleges have become more attractive to students. Currently, IT certifications are valued assets for individuals seeking employment or advancement in the computing field. Furthermore, graduates with key IT certificates have the opportunity to be readily employable and be effective in their work.

8. References

1. Criteria for Accrediting Computing Programs, Computing Accreditation Commission.
http://www.abet.org/criteria_cac.html
<http://www.abet.org/images/Criteria/C001%2004-05%20CAC%20Criteria%2011-18-03.pdf>
2. Gorgone, J., Davis, G., Valacich, J. , Topi, H., Feinstein, D. & Longenecker, H., IS 2002: Model curriculum and guidelines for undergraduate programs in information systems, 2002.
<http://www.acm.org/education/is2002.pdf>
3. 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, 2004.
http://www.asee.org/acPapers/2004-1708_Final.pdf
4. Dean, T., Network+ Guide to Networks, 3rd edition, Course Technology, 2004
5. Grice, M., Lab Manual for Network+ Guide to Networks, 3rd Edition, Course Technology, 2004.
<http://www.course.com/networking/>
6. Hudson, K., Caudle, K., and Cannon. K, CCNA Guide to Cisco Networking, 2nd edition, Course Technology, 2003.
7. Cannon, K., Lab Manual for CCNA Guide to Cisco Networking, 2nd edition, Course Technology, 2003.
8. Mughal, K. and Rasmussen, R., A Programmer's Guide to Java Certification, Addison-Wesley., 2000.
9. Savitch, W., JAVA: An Introduction to Computer Science and Programming, 3rd edition, Prentice Hall, 2004.
10. Deitel, H., & Deitel, P., Java How to Program, 5th Edition, Prentice-Hall, 2003.

Biography

AKRAM AL-RAWI

Akram Al-Rawi is a Sun certified Java Programmer and a Professor of CIS 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.

AZZEDINE LANSARI

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. Since 1998, he is an Assistant Professor at Zayed University. 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

Faouzi Bouslama is a Professor at Zayed University. He received a Ph D degree in Electronic Engineering from Shizuoka University, Japan 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. He joined Zayed University in August 2000. His research interests include Neural Networks, Fuzzy Logic, and Curriculum development.

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.