



EUCIP
European Certification of
Informatics Professionals

Core Syllabus

**B – “BUILD” KNOWLEDGE AREA:
DEVELOPMENT AND IMPLEMENTATION OF INFORMATION SYSTEMS**

Version 2.6

June 2006

EUCIP CORE Version 2.6 Syllabus.

The following is the Syllabus for *EUCIP CORE Version 2.6*, which provides the basis for the tests in this module domain.

Module Goals

EUCIP CORE:
Build Knowledge Area Module B, Build, encompasses the traditional technical aspects of design, specification, development, testing, integration and deployment of IT systems. It requires the candidate to understand the systems development lifecycle, to know about the typical development process and be aware of recent software development trends. The Candidate shall also be able to apply relevant methodologies and tools. He / She shall appreciate the design principles associated with user interfaces, web pages and hypermedia, as well as those used in relational databases and data warehouses. The candidate will also gain a rudimentary knowledge of query languages and be aware of some important database administration issues. In addition, the candidate shall be able to apply typical data structures and algorithms, to recognise different programming languages and constructs, to evaluate the issues in maintaining, and know about documenting and testing software systems.

B – “BUILD” KNOWLEDGE AREA: DEVELOPMENT AND IMPLEMENTATION OF INFORMATION SYSTEMS

Category	Topic	Ref	Item
B.1 Systems Development Process and Methods	B.1.1 Application Software and System Software	B.1.1.1	Recognise and describe the difference between systems software and application software
		B.1.1.2	Name some examples of both categories
		B.1.1.3	Explain the use of application and systems software
	B.1.2 Systems Development Principles and Methodologies	B.1.2.1	Understand and describe the basic steps of Systems Development
		B.1.2.2	Differentiate between and give examples of different models of Systems Development
		B.1.2.3	Describe the life-cycle of a system
	B.1.3 Systems Development Tools	B.1.3.1	Explain the use of tools (e.g. “Upper”, “Lower” and Integrated CASE tools) at different stages of Systems Development

Category	Topic	Ref	Item
		B.1.3.2	Understand and describe the strengths and weaknesses of different development tools and techniques
		B.1.3.3	Demonstrate use of a simple development tool
	B.1.4 Software and System Testing	B.1.4.1	Describe the different types of testing and review applied during the systems lifecycle (e.g. as defined in the V-model)
	B.1.5 System Implementation	B.1.5.1	Explain the use of tools at systems implementation (i.e. the phase within a System Development lifecycle incorporating the release of software to users, training users and providing initial support for users) and the different steps in using one of the tools
		B.1.5.2	Understand and describe the strengths and weaknesses of different implementation tools
	B.1.6 System Control and Safety	B.1.6.1	Describe in outline the different types of systems which can protect company-sensitive data (e.g. physical, procedural).
		B.1.6.2	Describe the day-to-day security routines in a distributed system (e.g. back-up procedures, access control).
	B.1.7 Trends in Systems Development	B.1.7.1	Describe different Systems Development approaches
		B.1.7.2	Describe the major technical architectures for which systems are being developed (e.g. two-tier client server variants, three-tier client server, "n" tier web-based, legacy mainframe extension and integration)
B.2 Data Management and Databases	B.2.1 Data and Transactions	B.2.1.1	Understand the use of data in a database system
		B.2.1.2	Understand the advantages and the problems of storing data and information, in relation to: data redundancy and inconsistency, integrity problems, flexibility in data, concurrent access and security
		B.2.1.3	Understand the difference between data and information
	B.2.2 Data Modelling	B.2.2.1	Know about data abstraction, physical level, conceptual level and view level
		B.2.2.2	Know about different groups of data models: object-based logical model , record-based logical model and physical data model

Category	Topic	Ref	Item
		B.2.2.3	Know the principles of object-based logical models: i.e. the entity-relationship model, the object-oriented model
		B.2.2.4	Know the difference between the record-based logical models: network model and hierarchical model
	B.2.3 Files and Databases	B.2.3.1	Explain the difference between a file management system and a Database Management System
		B.2.3.2	Describe the components of a database system, including: data files, data dictionary, indices and statistical data
		B.2.3.3	Know about file organisation and the advantage/disadvantage with files based on fixed length records and variable length records
	B.2.4 Database Management Systems	B.2.4.1	Explain the business areas where Database Management Systems are used, and why
		B.2.4.2	Explain the components of a Database Management System: programs and types of data
		B.2.4.3	Describe in general terms: Data Definition Language (DDL) Data Manipulation Language (DML), Database Manager, Database Administrator (DBA), database users
	B.2.5 Data Warehousing and Data Mining	B.2.5.1	Explain the concept and components of a Data Warehousing system
		B.2.5.2	Describe the functions of data mining
		B.2.5.3	Describe where Data Warehousing systems are mostly used
	B.2.6 The Relational Model	B.2.6.1	Describe the benefits of a relational model: redundancy etc.
		B.2.6.2	Be able to explain simple examples of the process of normalisation for 1st. normal form to 3rd. normal form
	B.2.7 Queries and Reports	B.2.7.1	Know the difference between procedural and non-procedural query languages
		B.2.7.2	Describe the fundamental operations of the relational algebra
		B.2.7.3	Describe the different components of a SQL language: DDL, DML, View etc.

Category	Topic	Ref	Item
		B.2.7.4	Use and demonstrate SQL operations: where, from, select etc.
	B.2.8 Database Administration	B.2.8.1	Know the definition and function of a Database Management System
		B.2.8.2	Know the functions of a Database Administrator (DBA)
		B.2.8.3	Explain the contents of some of the most important database administration procedures: schema definition, storage structure and access methods, schema and physical organisation modification, authorisation for data access, etc.
	B.2.9 Security and Integrity of Data	B.2.9.1	Explain in general the different security and integrity problems represented by the acronym CIA (confidentiality, integrity, availability): e.g. integrity constraints, accidental loss of data integrity, accidental loss of data consistency and intentional (malicious) access to the database
		B.2.9.2	Give examples of different security items grouped as: human security, physical security, Operating System security and database security
		B.2.9.3	Describe recovery schemes based on different types of failure like: logical errors, system errors, system crash and disk failure.
B.3 Programming	B.3.1 Software Design Methods and Techniques	B.3.1.1	Know different programming design methods, such as Object-Oriented (OO) design, "top down" design, structured programming
		B.3.1.2	Know how to use abstraction as a technique of problem-solving and design
		B.3.1.3	Understand the specific needs of legacy systems in program design
	B.3.2 Data Structures and Algorithms	B.3.2.1	Understand different data structures such as records, arrays, and linked lists
		B.3.2.2	Understand the relationship between different widely used algorithms and the data structures above
	B.3.3 Types of Programming Languages	B.3.3.1	Know the main types of programming languages (different generations, functional, procedural, OO-based)
		B.3.3.2	Understand the role of syntax in programming languages
		B.3.3.3	Understand the difference between compilation and interpretation of programming languages

Category	Topic	Ref	Item
	B.3.4 Introduction to Programming Concepts	B.3.4.1	Understand the most important constructs of the programming language i.e.: i. Data Definitions ii. Input/Output (IO) iii. Control Statements iv. Arithmetic and logical operations v. Subroutines
	B.3.5 Testing	B.3.5.1	Understand the main functions of automatic test tools
		B.3.5.2	Demonstrate knowledge of test methodologies
		B.3.5.3	Understand the main features of module, link and system testing
	B.3.6 Documentation	B.3.6.1	Demonstrate knowledge of basic documentation requirements for software deliverables and development (e.g. Structured English, decision trees)
		B.3.6.2	Understand the concept of well-structured programs and related documentation
	B.3.7 Maintenance	B.3.7.1	Know how to document changes in software and program documentation
		B.3.7.2	Give examples of methods that can be used to attain quality in program maintenance, e.g. code inspections
	B.3.8 Programming Examples	B.3.8.1	Understand small program segments constructed from a given hypothesis
B.4 User Interface and Web Design	B.4.1 Human-computer Interaction	B.4.1.1	Understand common communications theory: sender, messages, receiver
		B.4.1.2	Understand how to communicate information
	B.4.2 Graphic design	B.4.2.1	Use basic graphic principles, based on normal written text e.g. font size, percentage white space
		B.4.2.2	Use basic graphic principles for illustrations and colours
		B.4.2.3	Use standard picture manipulation
	B.4.3 Current Methods and Techniques	B.4.3.1	Describe the tools for development of a web-site
		B.4.3.2	Construct structure diagram for design of web-site
		B.4.3.3	Describe the major methods for navigation

Category	Topic	Ref	Item
		B.4.3.4	Know the principles of how to use story boards, rough drafts etc.
B.4.4	Guidelines and Standards for User Interfaces	B.4.4.1	Describe tools/guidelines for developing user-friendly web-sites
		B.4.4.2	Define "best practice" for good web text
		B.4.4.3	Define test models for testing of user interface requirements/goals
B.4.5	Characteristics of the Web, its Possibilities and Constraints	B.4.5.1	Know the history behind the world-wide-web
		B.4.5.2	Explain the traditional components of a web-site
		B.4.5.3	Explain the use of web-sites in a company, internal and external
		B.4.5.4	Describe models for analysing and revealing the need within the target group
		B.4.5.5	Be aware of the disadvantages of web-sites, giving examples
B.4.6	Hypertext and Hypermedia	B.4.6.1	Describe the use of hypertext and hypermedia
		B.4.6.2	Use hypertext and hypermedia
B.4.7	Central Problems in Web Design	B.4.7.1	Understand the needs of the target group for the web-site
		B.4.7.2	Understand the risks of having too many message on one page
		B.4.7.3	Understand the problems around a poor mix of colours
		B.4.7.4	Understand the need for easy navigation on the site
B.4.8	Designing Web Pages	B.4.8.1	Be able to use the basic commands in HTML
		B.4.8.2	Understand the main areas of HTML (hard format, soft format, special characters, dividers, alignment, headers, image-tags, backgrounds, colours, links, lists, tables, forms and frames)
		B.4.8.3	Understand the concept of "style sheets" and their use in design