

Syllabus EUCIP Core – BUILD - 3.1

[*Vedi nota relativa al copyright*](#)

CATEGORY	TOPIC	REF.	ITEM	
B.1 Systems Development Process and Methods	B.1.1 Software in Data Processing Systems	B.1.1.1	Describe a data processing system as a combination of hardware, firmware, operating system software, application software, system configuration data and user-defined data.	
		B.1.1.2	Recognise and list examples of system software.	
		B.1.1.3	Recognise and list examples of application software.	
	B.1.2 Systems Development Life Cycles	B.1.2.1	Describe the typical phases of systems development.	
		B.1.2.2	Compare the various classical models of systems development life cycle, such as waterfall, spiral, prototyping, incremental releases.	
		B.1.2.3	Describe the life cycle of a system in terms of analysis, development and deployment, use and maintenance, decommissioning.	
		B.1.2.4	Outline specifications for requirements and design, such as organisational specification, technical specification.	
	B.1.3 Software Development Tools	B.1.3.1	Outline the use of tools at different stages of systems development, such as upper, lower, integrated CASE tools.	
	B.1.4 System Testing and Deployment	B.1.4.1	Describe the different types of testing and review that can be applied during the systems development life cycle.	
		B.1.4.2	Describe the main issues in the system implementation phase (deployment), such as release of software to users, data migration, user training and initial support.	
	B.1.5 System Control and Safety	B.1.5.1	Distinguish between development, test and production environments and understand the importance of a structured approach to system releases, such as version control systems, software distribution procedures.	
	B.2 Data Management and Databases	B.2.1 Data and Transactions	B.2.1.1	Outline the importance of durable data recording for transaction processing and reporting systems.
			B.2.1.2	Describe how atomicity, consistency, isolation and durability help to guarantee secure database transactions.
B.2.2 Database Structure		B.2.2.1	Distinguish between a file management system and a database management system (DBMS).	
		B.2.2.2	Describe the components of a database system, such as data files, data dictionary, indexes, statistical data.	
		B.2.2.3	Outline the business areas where a DBMS is used, and recognise the benefits they offer.	
		B.2.2.4	Outline the components of a DBMS, such as query language, report generator, administration tools, concurrency controls, transaction management, backup/recovery tools.	
		B.2.2.5	Describe the different roles of a database administrator, database designer/programmer, database user.	
B.2.3 Data Modeling		B.2.3.1	Define data abstraction and describe the difference between physical level, conceptual (logical) level, view (user) level.	
B.2.4 The Relational Model		B.2.4.1	Outline the benefits of a relational model, such as non-redundancy, flexibility, scalability.	
		B.2.4.2	Describe important relational model terms, such as relation, key, primary key, alternate key, foreign key, referential integrity.	
B.2.5 Query Languages		B.2.5.1	Distinguish between procedural and non-procedural query languages.	
		B.2.5.2	Describe the fundamental operations of relational algebra, such as select, project, rename, cartesian product, union, joins, set difference.	
		B.2.5.3	Describe the components of the Structured Query Language (SQL), such as Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL).	
		B.2.5.4	Understand SQL DDL commands, such as create, drop, alter table.	
		B.2.5.5	Understand SQL DCL commands, such as grant, revoke.	
B.2.6 SQL queries		B.2.6.1	Understand basic SQL DML commands, such as insert, delete, update, select.	
		B.2.6.2	Understand SQL clauses, such as where, order by, group by.	
		B.2.6.3	Outline the use of views and special SQL commands, such as commit, rollback.	
B.2.7 Database Administration and Security		B.2.7.1	Describe the most important database administration procedures, such as schema definition, storage structure and access methods, schema and physical organisation modification, authorisation for data access.	
		B.2.7.2	Describe the security and integrity problems covered by the acronym CIA (Confidentiality, Integrity, Availability), such as integrity constraints, accidental loss of data integrity,	
			accidental loss of data consistency, intentional (malicious) access to the database.	
		B.2.7.3	Outline examples of different security policies, such as human security,	

			physical security, operating system security, database security.
	B.2.8 Data Warehousing and Data Mining	B.2.8.1	Describe the concept and components of a data warehousing (DW) system.
		B.2.8.2	Define the concept of data mining.
		B.2.8.3	Recognise the principle uses of DW systems.
B.3 Programming	B.3.1 Software Design Methods and Techniques	B.3.1.1	Outline the main features of different program design methods, such as object-oriented (OO) design, top down design, structured programming.
		B.3.1.2	Describe the use of abstraction as a technique for problem-solving and software design.
		B.3.1.4	Distinguish between open source and proprietary software development.
		B.3.1.5	Outline the different licensing requirements of proprietary software, open source, free software, freeware.
	B.3.2 Data Structures and Algorithms	B.3.2.1	Describe structured and unstructured data types and recognise different data structures, such as records, arrays, linked lists.
		B.3.2.2	Evaluate the fit between typical search and sort algorithms and the different data structures.
	B.3.3 Programming Languages	B.3.3.1	Distinguish between and describe the merits of the main types of programming languages, such as functional, procedural, OO-based.
		B.3.3.2	Describe the use of procedures and functions, and distinguish between call by value and call by reference.
		B.3.3.3	Define the term syntax and outline its importance in programming languages.
		B.3.3.4	Distinguish between compilation and interpretation of programming languages.
	B.3.4 Object-Oriented Programming	B.3.4.1	Describe the main concepts of object-oriented design.
		B.3.4.2	Describe the concept of object-oriented programming.
		B.3.4.3	Describe the concept of a class, object, instance, method and their relationship in object-oriented programming.
		B.3.4.4	Describe the concept of inheritance and the impact of inheritance for the programmer.
		B.3.4.5	Describe the concepts of abstraction and encapsulation (information hiding).
		B.3.4.6	Describe how polymorphism contributes to efficient software design through the development of reusable components.
	B.3.5 Elementary Constructs	B.3.5.1	Interpret and evaluate input/output instructions.
		B.3.5.2	Interpret and evaluate control statements.
		B.3.5.3	Interpret and evaluate arithmetic and logical operations.
	B.3.6 Testing	B.3.6.1	Define basic testing concepts, such as error, fault, failure. Recognise the various levels of checking, testing, debugging.
		B.3.6.2	Describe the different purposes and scope of unit testing, system testing, acceptance testing.
		B.3.6.3	Distinguish between static and dynamic test methodologies, and list examples of automatic test tools.
	B.3.7 Documentation and Maintenance	B.3.7.1	Describe the common documentation for software development and delivery, such as structured English, decision trees, Unified Modeling Language (UML) code, code comments, flowcharts.
		B.3.7.2	Describe the value of well-structured and documented code.
B.3.7.3		Outline how to document changes in software and program documentation.	
B.3.7.4		Describe methods to attain quality in program maintenance, such as code inspections, rules for comments in the code, technical reference documents.	
B.3.8 Programming Examples	B.3.8.1	Interpret small program segments constructed from a given hypothesis.	
	B.3.8.2	Identify errors or weaknesses in the code and modify it to meet requirements.	
B.4 User Interface and Web Design	B.4.1 Human-Computer Interaction: Guidelines and Standards	B.4.1.1	Define basic concepts of communication theory, such as sender, messages, receiver.
		B.4.1.2	Understand how communication applies to human beings, and recognise effective ways to communicate information.
		B.4.1.3	Define the concept of a User Interface and list the different kinds of interfaces, such as textual, graphical, auditory.
		B.4.1.4	Outline the various technologies that communicate information to human senses, such as sound types, visual clues, digital smells, haptics/touch.
		B.4.1.5	List models for testing the effectiveness of a user interface against its design requirements and goals.
	B.4.2 Graphic Design	B.4.2.1	Outline the concepts of graphics and animation (bitmap and vector), digital audio and video, and describe their differences, uses, standard formats.
		B.4.2.2	Describe the benefits of using drawings, pictures, colours, animation, and apply basic principles of graphic design, such as balance, harmony, contrast, variety.

		B.4.2.3	Use common tools for simple picture manipulation in terms of size, shape, colours, contrast, transparency.
	B.4.3 Web and Hypermedia: Possibilities and Limitations	B.4.3.1	Describe the history behind the Internet and the World Wide Web.
		B.4.3.2	Define hypertext and hypermedia and outline their importance in web page design.
		B.4.3.3	Outline the common components used in web pages, such as top bar, side bar, site map, contact, search feature, help, last updated, navigation icons.
		B.4.3.4	Describe the use and value of internal and external websites in a company.
		B.4.3.5	Outline some of the challenges in maintaining a business website.
	B.4.4 Web Design Requirements and Methods	B.4.4.1	Recognise the needs of the target group for whom a web page has been designed.
		B.4.4.2	Outline the risks of having too many messages on one page.
		B.4.4.3	Outline the problems associated with a poor mix of colours. with a poor mix of colours.
		B.4.4.4	Describe guidelines for developing user-friendly websites, such as readability, prioritised content, easily navigable, consistently navigable, where am I.
		B.4.4.5	Describe general quality criteria for web text, such as browser capability issues, validating HTML, condense text content, spell-check, small byte-size graphics.
		B.4.4.6	Explain the need for easy navigation on a website.
		B.4.4.7	Outline the tools for the development of a website.
		B.4.4.8	Outline the use and purpose of structure diagrams for website design.
		B.4.4.9	Describe the major methods used for navigation.
		B.4.4.10	Describe some project approaches to web design and common techniques, such as story boards, rough drafts.
	B.4.5 Designing Web Pages	B.4.5.1	Outline the concept of a markup language and describe the main features of HTML.
		B.4.5.2	Use basic HTML commands and interpret layout commands, such as hard format, soft format, special characters, dividers, alignment, headers, image tags, backgrounds, colours, links, lists, tables, forms, frames.
		B.4.5.3	Outline basic graphic principles for normal written text, such as font size, percentage white space.
		B.4.5.4	Outline the basic elements of XML, its uses, the evolution of HTML into XHTML.
		B.4.5.5	Describe the concept of style sheets, such as Cascading Style Sheets (CSS) and Extensible Stylesheet Language (XSL). Understand their use in design.
	B.4.6 Web Based Programming	B.4.6.1	Distinguish between client-side and server-side technologies and recognise different types of web-based programming languages.
		B.4.6.2	List some of the main challenges in integrating web-based systems to existing systems.

Nota

I salti nella numerazione degli item (e in alcuni casi dei topic) derivano dal fatto che, per permettere un più agevole uso delle risorse esistenti basate sul Syllabus 3.0, si è mantenuta la numerazione di tale Syllabus.

© AICA-CEPIS – 2009. Tutti i diritti riservati

AICA e CEPIS informano i lettori di questo Syllabus tutti i diritti di proprietà intellettuale e industriale relativi al syllabus EUCIP core sono riservati a AICA e CEPIS. L'Utente ha la sola facoltà di lettura di tali documenti e, in via meramente esemplificativa e non esaustiva, non acquisisce alcun diritto di copia, e/o di utilizzo interno e/o esterno per eventuale commercializzazione.

AICA e CEPIS informano gli Utenti del Sito che tutti i diritti sulla metodologia e le procedure del proprio sistema sono protetti dalle normative vigenti e, pertanto, anche sotto il profilo di eventuali comportamenti di terzi che possano configurare concorrenza sleale e/o clonazione del proprio sistema.

Chi fosse interessato all'uso dei documenti, delle metodologie e del syllabus è invitato a rivolgersi ad AICA (info@eucip.it)