

<b>Classe delle lauree magistrali in:</b> Ingegneria Meccanica (L-9)		<b>Corso di laurea in:</b> Ingegneria Meccanica	<b>Anno accademico:</b> 2014 - 2015	
<b>Tipo di attività formativa:</b> Base	<b>Ambito disciplinare:</b> Ingegneria Informatica	<b>Settore scientifico disciplinare:</b> Sistemi di Elaborazione delle Informazioni (ING-INF/05)	<b>CFU:</b> 6	
<b>Titolo dell'insegnamento:</b> Fondamento di Informatica (supplenza)	<b>Codice dell'insegnamento:</b> 2011	<b>Tipo di insegnamento:</b> Obbligatorio	<b>Anno:</b> primo	<b>Semestre:</b> secondo
<b>DOCENTE:</b> Prof. Vitoantonio Bevilacqua (RU)				
<b>ARTICOLAZIONE IN TIPOLOGIE DIDATTICHE:</b> 48 ore di lezioni teoriche (6 CFU)				
<b>PREREQUISITI:</b> Conoscenze elementari di aritmetica e teoria degli insiemi.				
<b>OBIETTIVI FORMATIVI:</b> Scopo dell'insegnamento è quello di spiegare la aritmetica binaria, la logica booleana, la architettura del calcolatore, le strutture dati, la rappresentazione degli algoritmi tramite diagrammi di flusso (flow-chart) e la modellazione di scenari basata su principi di ingegneria del software (Unified Modelling Language: UML), le strutture di controllo della programmazione strutturata, la progettazione, la implementazione e il testing di algoritmi scritti in linguaggio C.				
<b>CONTENUTI:</b> <b>Codifica binaria delle informazioni e algebra di boole:</b> Codifica di numeri interi e razionali. Il concetto di precisione di una rappresentazione ed il calcolo dell'errore relativo. Algebra di Boole Il concetto di variabile, funzione booleana e tavola della verità. Gli operatori logici fondamentali. Minimizzazione con metodo mappe di Karnaugh. Codice di rilevazione e correzione di errori (Hamming) <b>Architettura dei calcolatori</b> L'architettura del calcolatore di Von Neumann: memoria centrale; processore; bus; interfacce I/O; periferiche; le memorie di massa. Il linguaggio macchina: formato delle istruzioni; cenni ai metodi di indirizzamento. Linguaggi simbolici: cenni sul linguaggio assembler. La memoria virtuale paginata. <b>Algoritmi e strutture dati</b> Definizione di algoritmo; proprietà di un algoritmo. Teorema di Böhm-Jacopini. Algoritmi strutturati: strutture di controllo fondamentali, sequenza, selezione, iterazione. Concetto di complessità computazionale di un algoritmo. La ricorsione: principio di induzione, definizione di problema ricorsivo, esempi. Linguaggi per la descrizione di algoritmi, il diagramma di flusso (flow chart) . <b>Compilatori ed interpreti:</b> analisi lessicale, sintattica, semantica e generazione del codice oggetto <b>Programmazione strutturata:</b> Linguaggi di programmazione: definizione di un linguaggio: alfabeto dei simboli, sintassi, semantica. <b>La struttura dei programmi in C:</b> parte dichiarativa e parte esecutiva. Le variabili di un programma. I tipi di dati: definizione di tipo di dato; tipi di dati predefiniti ( built-in) e tipi definiti dall'utente (user-defined); tipi semplici: int, float, char e double. <b>Strutture di controllo in C:</b> if- else, while; altre istruzioni cicliche: for, while, do while; costruito switch – case. <b>Tipi strutturati:</b> caratteristiche delle variabili array e strutture; costruttori di tipo: il costruttore array, il costruttore struct, il costruttore di puntatori. <b>Le funzioni:</b> definizione di una funzione; dichiarazione di funzione: la testata (header); l'istruzione return; i parametri: passaggio dei parametri; chiamata della funzione; parametri attuali e parametri formali; gestione della memoria nella chiamata di funzioni, record di attivazione. Regole di visibilità delle variabili. Funzioni predefinite, lettura e scrittura; librerie di funzioni. <b>La gestione dei file di testo.</b> definizione; la variabile puntatore a file; operazione di gestione di file: apertura (modalità di utilizzo), chiusura, scrittura, lettura. <b>Introduzione alla programmazione orientata agli oggetti e il C++.</b>				
<b>METODI DI INSEGNAMENTO:</b> Lezioni teoriche in aula e analisi di codice in linguaggio C.				
<b>CONOSCENZE E ABILITÀ ATTESE:</b> Capacità di rappresentare algoritmi attraverso diagrammi di flusso (flow-chart); Capacità di progettare, implementare e testare algoritmi in programmazione strutturata (linguaggio C)				
<b>SUPPORTI ALLA DIDATTICA:</b> PC con ambiente di sviluppo per C/C++ e videoproiettore. Appunti dalle lezioni, dispense del docente, calendario degli esami e avvisi dal sito del docente <a href="http://www.vitoantoniobevillacqua.it/">http://www.vitoantoniobevillacqua.it/</a> e ambienti di file sharing				
<b>CONTROLLO DELL'APPRENDIMENTO E MODALITÀ D'ESAME:</b> Esame scritto volto alla verifica della conoscenza del linguaggio C e successivo orale, comprendente la discussione di un progetto d'anno di implementazione di uno scenario reale				
<b>TESTI DI RIFERIMENTO PRINCIPALI:</b> Linguaggio C - Guida alla programmazione 4/ed Alessandro Bellini, Andrea Guidi C++: The Complete Reference, 4th Edition - Herbert Schildt C++ Fondamenti di programmazione - Harvey M. Deitel, Paul J. Deitel				
<b>ULTERIORI TESTI SUGGERITI:</b> C/C++: Programmer's Reference Herbert Schildt				
<b>ALTRE INFORMAZIONI:</b> Dipartimento di Ingegneria Elettrica e dell'Informazione, Politecnico di Bari ( <a href="http://dee.poliba.it">http://dee.poliba.it</a> ) Stanza docente 3° piano, tel. 0805963326 (int 3326), e-mail: <a href="mailto:vitoantonio.bevillacqua@poliba.it">vitoantonio.bevillacqua@poliba.it</a> .				

<b>Master Degree class:</b> Mechanical Engineering (L-9)		<b>First level (three years) degree:</b> Mechanical Engineering (L-9)		<b>Academic year:</b> 2014 - 2015	
<b>Type of course</b> Basis	<b>Disciplinary area:</b> Information Engineering	<b>Scientific Discipline Sector:</b> (ING-INF/05)		<b>ECTS Credits:</b> 6	
<b>Title of the course:</b> Fundamental of Informatics	<b>Code:</b> 2011	<b>Type of course:</b> DUE		<b>Year:</b> 1 <sup>st</sup>	<b>Semester:</b> 2 <sup>nd</sup>
<b>LECTURER:</b> Prof. Vitoantonio Bevilacqua (Tenured Assistant Professor)					
<b>Hours of Instruction:</b> 48 hours of theory (6 ECTS)					
<b>PREREQUISITES:</b> Basic knowledge in arithmetic and set theory					
<b>AIMS:</b> The aim of this course is to explain the binary arithmetic, the boolean algebra, the computer architecture, the data structures, algorithms representation through flow charts and scenarios modellation based on software engineering principle (Unified Model Language: UML), control structures of structured programming, the characteristic of the object oriented programming (OOP), the design, the implementation and the testing of algorithms written in C language.					
<b>CONTENTS:</b> <b>Binary coding of information and boolean algebra:</b> Coding of integer and rational numbers. The concept of the precision of the representation and calculation of the relative error. Boolean algebra, the variable concept, boolean function and true table. Basic logic operators. Minimization through Karnaugh map. Error detection and correction (Hamming code). <b>Computer architecture</b> Von Neumann architecture: central memory, processor, bus, I/O interfaces, peripherals, mass storage. The machine code: instruction format, mention of addressing mode. Symbolic languages: mention on assembly language. Virtual memory and paging. <b>Algorithms and data structures</b> Definition of the algorithm; algorithm properties. Böhm-Jacopini theorem. Structured algorithm: basic control structure, sequence, selection and iteration. Algorithm computational complexity concept. The recursion: induction principle, definition of the recursive problem and examples. Language for the algorithm description, flow chart. <b>Compilers and interpreters:</b> lexical, syntax and semantic analysis, generation of the object code. <b>Structured programming:</b> programming languages: definition of a language: alphabet, syntax, semantic. <b>Programs structure in C:</b> declarative and executive section. Variables of a program. Data types: data type definition, predefined types (built-in) and user-defined types; simple types: int, float, char and double. <b>Control structures in C:</b> if-else, while; other cyclic instruction: for, do-while, switch-case. <b>Structured types:</b> array and structured variables characteristics; type constructor: array constructor, struct constructor, pointers constructor. <b>Functions:</b> function definition, declaration of a function: header, return instruction, parameters: passing parameters, call of a function, current and formal parameters, memory management, activation record. Visibility rules of a variable. Read, write and default functions; libraries. <b>File and memory management:</b> definition, file pointer, text files, file management operation: open, close, write, read. <b>Introduction to Object oriented programming and C++</b>					
<b>TEACHING METHODS:</b> Theoretical lectures in classroom and code analysis in C language.					
<b>EXPECTED OUTCOME AND SKILLS:</b> Ability to represent algorithms through flow charts; Ability to design, implement and test algorithms in structured programming (C language)					
<b>TEACHING AIDS:</b> Laptop with development environment for C/C++ languages and video projector. Lecture notes, teacher notes, exams calendar and news from <a href="http://www.vitobevilacqua.it/">http://www.vitobevilacqua.it/</a> and file sharing systems.					
<b>EXAMINATION METHOD:</b> Oral examination to verify all the knowledge and C language coding skills, including discussion about a project implemented on a real scenario.					
<b>BIBLIOGRAPHY:</b> Linguaggio C - Guida alla programmazione 4/ed Alessandro Bellini, Andrea Guidi C++: The Complete Reference, 4th Edition - Herbert Schildt C++ Fondamenti di programmazione - Harvey M. Deitel, Paul J. Deitel					
<b>FURTHER BIBLIOGRAPHY:</b> C/C++: Programmer's Reference Herbert Schildt					
<b>FURTHER INFORMATIONS:</b> Department of Electrical and Information Engineering, Politecnico di Bari ( <a href="http://dee.poliba.it">http://dee.poliba.it</a> ) Lecturer room at 3rd floor, tel. 0805963326 (int 3326), e-mail: <a href="mailto:vitoantonio.bevilacqua@poliba.it">vitoantonio.bevilacqua@poliba.it</a> .					