



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Matematica e Informatica
ANNO ACCADEMICO OFFERTA	2020/2021
ANNO ACCADEMICO EROGAZIONE	2021/2022
CORSO DILAUREA	INFORMATICA
INSEGNAMENTO	LINGUAGGI DI PROGRAMMAZIONE
TIPO DI ATTIVITA'	B
AMBITO	50166-Discipline Informatiche
CODICE INSEGNAMENTO	04758
SETTORI SCIENTIFICO-DISCIPLINARI	INF/01
DOCENTE RESPONSABILE	FICI GABRIELE Professore Associato Univ. di PALERMO
ALTRI DOCENTI	
CFU	9
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	153
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	72
PROPEDEUTICITA'	05880 - PROGRAMMAZIONE E LABORATORIO C.I.
MUTUAZIONI	
ANNO DI CORSO	2
PERIODO DELLE LEZIONI	1° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	FICI GABRIELE Martedì 15:00 16:00 Studio 202, DMI. Giovedì 15:00 16:00 Studio 202, DMI.

DOCENTE: Prof. GABRIELE FICI

PREREQUISITI	Conoscenze di programmazione in C, matematica discreta (relazioni, calcolo combinatorio), conoscenze di base di logica (predicati, tecniche di dimostrazione, principio di induzione).
RISULTATI DI APPRENDIMENTO ATTESI	<p>Conoscenza e capacita' di comprensione Conoscenza panoramica dei vari paradigmi di programmazione. Conoscenza approfondita del paradigma orientato agli oggetti e del paradigma funzionale. Conoscenza approfondita del linguaggio Java Standard Edition nelle sue ultimissime versioni.</p> <p>Capacita' di applicare conoscenza e comprensione Capacita' di valutare le funzionalita' dei diversi paradigmi di programmazione. Capacita' di scrivere codice anche avanzato nel linguaggio di programmazione Java.</p> <p>Autonomia di giudizio Capacita' di valutare e comparare autonomamente le soluzioni di un problema di limitata complessita'.</p> <p>Abilita' comunicative Capacita' di organizzarsi in gruppi di lavoro. Capacita' di comunicare efficacemente in forma orale, anche utilizzando termini in inglese.</p> <p>Capacita' di apprendere Capacita' di catalogare, schematizzare e rielaborare le nozioni acquisite.</p>
VALUTAZIONE DELL'APPRENDIMENTO	<p>L'esame finale consiste in una prova scritta e una prova orale. La prova scritta e' costituita da un test a risposta multipla di tipo vero/falso (5 opzioni di risposta per ogni domanda, una sola risposta esatta), costituito da 30 domande. Ogni risposta esatta vale un punto, ogni risposta sbagliata vale -0.25 punti, ogni risposta non fornita vale 0 punti. La prova orale consiste in un colloquio che verte ad approfondire le conoscenze acquisite e la capacita' di scrivere codice corretto (su carta).</p> <p>Concorrono a determinare il voto: 1) Il risultato della prova scritta; 2) La chiarezza e correttezza delle risposte al colloquio orale; 3) La chiarezza e correttezza del codice scritto su carta durante il colloquio orale.</p> <p>Fasce di valutazione: Il voto va da 18/30 (prova scritta sufficiente, conoscenze minime sul programma, valutate in base alle risposte fornite alle domande rivolte al colloquio orale, capacita' di scrivere codice elementare) a 30/30, con eventuale lode (prova scritta eccellente, ottime conoscenze sul programma, valutate in base alle risposte fornite alle domande rivolte al colloquio orale, ottime capacita' di scrivere codice anche complesso).</p>
OBIETTIVI FORMATIVI	Fornire competenze di base e avanzate, sia metodologiche che tecniche, sui linguaggi di programmazione, in particolar modo sui linguaggi orientati agli oggetti e sui linguaggi funzionali.
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali in laboratorio. Gli studenti avranno a disposizione dei computer per seguire le lezioni in modo interattivo. La frequenza delle lezioni e' fortemente raccomandata.
TESTI CONSIGLIATI	<p>1) C. De Sio Cesari, Manuale di Java 9, Hoepli (per la parte su Java - for the part on Java);</p> <p>2) C. Horstmann, Concetti di informatica e fondamenti di Java, Apogeo; (per la parte su Java - for the part on Java);</p> <p>3) Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft, Modern Java in Action: Lambdas, Streams, Functional and Reactive Programming (per la parte su programmazione funzionale - for the part on functional programming)</p>

PROGRAMMA

ORE	Lezioni
2	Introduzione e Primi Passi in Java: Sintassi di base - Primo programma - Commenti al codice - Letterali - Dichiarazione di variabili - Identificatori - Tipi primitivi: byte, short, int, long, float, double, boolean, char - Operatori unari, binari, condizionali e di concatenazione
3	Le classi Java I: Classi e oggetti - Attributi e metodi - Convenzioni su maiuscole e minuscole - Il metodo main - Costruttori - L'operatore new - Il costruttore di default - Riferimenti - Confronto di riferimenti - Variabili di istanza e variabili locali - Accesso ad attributi e metodi delle classi con dot notation - La Javadoc.
3	Le classi Java II: I metodi print, println e printf - Incapsulamento dei dati - Interfaccia pubblica e interfaccia privata - Modificatori di visibilita' public e private - Il riferimento implicito this - Deallocazione automatica della memoria e Garbage Collector - Firma di un metodo - Overloading di metodi - Overloading di costruttori - Invocazione di altri costruttori mediante "this()" - Variabili final
3	Le classi Java III: Attributi static - Metodi static - Organizzazione e compilazione di un package - Visibilita' package-private - Creazione di file jar eseguibili
2	Controllo del flusso: Costrutti di selezione, iterazione e salto: if, switch, while, do-while, for, break, continue, return - Il for esteso

PROGRAMMA

ORE	Lezioni
6	Stringhe e Wrapper: Il tipo String - Istanziamento di stringhe - Il pool di stringhe - La codifica Unicode per i caratteri - L'operatore di concatenazione - Il metodo equals per le stringhe - Il metodo compareTo per le stringhe - I metodi per le stringhe: startsWith e endsWith, length, substring, contains - toLowerCase e toUpperCase, trim, charAt, indexOf, lastIndexOf, replaceFirst, replaceAll, valueOf - Espressioni regolari - I metodi che usano le espressioni regolari: matches, replaceFirst, replaceAll, split - La classe StringBuilder - I metodi append, insert, delete, deleteCharAt, replace, reverse - La classe Scanner e i suoi metodi: next, nextLine, nextInt, nextDouble - Le classi wrapper - Boxing e unboxing - I metodi isDigit e isLetter della classe Character - I metodi parseInt, parseShort, parseFloat - I metodi toBinaryString, toOctalString e toHexString
4	Array e ArrayList: Creazione e inizializzazione di un array - Array di riferimenti a oggetti - L'attributo length - Il for esteso sugli array - I metodi della classe Arrays - La classe ArrayList - Cenni su Generics e classi parametriche - I metodi per gli ArrayList: add, size, get, set, remove, indexOf, lastIndexOf, contains, toArray - I metodi della classe Collections
3	Metodi Universali: I metodi presenti in tutte le classi - I metodi equals, toString e hashCode - Overriding del metodo toString - Overriding del metodo equals - Funzioni hash - Collisioni e hashing perfetto - Bucket e tabelle hash - Overriding del metodo hashCode - Mutua compatibilita' di questi overriding.
4	Ereditarieta' I: Riutilizzo e distribuzione dei programmi Java - Sottoclassi e superclassi - La parola chiave extends - Ereditarieta' di attributi e metodi - Il modificatore protected - Overriding di metodi - L'annotazione @Override - Attributi adombrati (shadowing) - Classi e metodi final
4	Ereditarieta' II: Ereditarieta' e costruttori - Invocare costruttori della superclasse con "super()" - Il riferimento super - Ereditarieta' e membri static - Ereditarieta' singola e Diamond Problem - Subtyping
4	Polimorfismo: Type checking statico - Binding dinamico - Definizione di polimorfismo - polimorfismo per metodi e polimorfismo per dati - array polimorfi - metodi virtuali e attributi adombrati - Il costrutto instanceof
6	Classi astratte e interfacce: Il modificatore abstract - Classi astratte e sottoclassi concrete - Classi astratte e membri static - Definizione e dichiarazione di un'interfaccia - Contenuto di un'interfaccia - Implementare un'interfaccia - Conflitto di nomi di variabili tra un'interfaccia e una classe che la implementa - Implementazioni parziali di interfacce in classi astratte - Estensione di interfacce - Ereditarieta' multipla per interfacce - Diamond problem, shadowing, overriding e overloading nell'ereditarieta' multipla per interfacce - Riferimenti a un'interfaccia - Interfacce ed ereditarieta' - Interfacce e polimorfismo - Focus sull'interfaccia Comparable - Comparatori e interfaccia Comparator - Implementazione di comparatori - Classi interne e classi interne anonime - Implementazione di un'interfaccia in una classe interna anonima - Tipi enumerativi
6	Input/output: Flussi (stream) di dati - Dati in ingresso e dati in uscita - Flussi di caratteri e flussi di byte - La classe File - I metodi della classe File: delete, exists, getAbsolutePath, getName, getPath, isDirectory, isFile, length, list, mkdir e renameTo - Usare Scanner per leggere dati da un file testuale - I metodi hasNext e hasNextLine, next e nextLine - Gestione di flussi di byte: le classi astratte InputStream e OutputStream e le loro sottoclassi FileInputStream e FileOutputStream - I metodi read e write per FileInputStream e FileOutputStream - Gestione di flussi di caratteri: le classi astratte Reader e Writer e le loro sottoclassi - I metodi read e write per FileReader e FileWriter - Bufferizzazione: le classi BufferedReader e BufferedWriter - La classe PrintWriter Il metodo printf - Filtraggio di classi - Flussi di oggetti - Le classi ObjectInputStream e ObjectOutputStream - Serializzazione e deserializzazione - L'interfaccia Serializable - I metodi readObject e writeObject - Serializzazione e membri static - Il modificatore transient - Memorizzazione dei dati di un programma su file - Gestione delle eccezioni - Eccezioni controllate e non - Il costrutto try/catch - Il try with resources
6	Programmazione generica - Tipi parametrizzati - Collezioni - Java Collection Framework - Le interfacce Iterable, Collection, Set e List - Le classi HashSet, TreeSet, ArrayList e LinkedList - Le interfacce Map e SortedMap - Le classi HashMap e TreeMap
4	Programmare un'interfaccia grafica per un'applicazione Java - AWT e Swing - Delegation model e gestione degli eventi - Ascoltatori - I componenti di Swing
6	Interfacce funzionali - Espressioni lambda - Method references - Stream - La streaming API di Java - Operazioni intermedie e terminali - Operazioni di aggregazione e di riduzione - La classe Collectors
6	Il paradigma di programmazione funzionale - Cenni storici - Funzioni di ordine superiore - Currying - Chiusure - Trasparenza referenziale (determinismo, assenza di effetti collaterali, assenza di mutazioni di stato) - Valutazione lazy e valutazione eager - Java e la programmazione funzionale - Stile funzionale e programmazione ricorsiva in Java - Programmazione funzionale pura in Java: tecniche e limiti