



# UNIVERSITÀ DEGLI STUDI DI PALERMO

<b>DIPARTIMENTO</b>	Ingegneria
<b>ANNO ACCADEMICO OFFERTA</b>	2017/2018
<b>ANNO ACCADEMICO EROGAZIONE</b>	2018/2019
<b>CORSO DILAUREA</b>	INGEGNERIA DELL'INNOVAZIONE PER LE IMPRESE DIGITALI
<b>INSEGNAMENTO</b>	PROGRAMMAZIONE
<b>TIPO DI ATTIVITA'</b>	B
<b>AMBITO</b>	50289-Ingegneria informatica
<b>CODICE INSEGNAMENTO</b>	05871
<b>SETTORI SCIENTIFICO-DISCIPLINARI</b>	ING-INF/05
<b>DOCENTE RESPONSABILE</b>	LO PRESTI LILIANA      Professore Associato      Univ. di PALERMO
<b>ALTRI DOCENTI</b>	
<b>CFU</b>	9
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	144
<b>NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA</b>	81
<b>PROPEDEUTICITA'</b>	
<b>MUTUAZIONI</b>	
<b>ANNO DI CORSO</b>	2
<b>PERIODO DELLE LEZIONI</b>	1° semestre
<b>MODALITA' DI FREQUENZA</b>	Facoltativa
<b>TIPO DI VALUTAZIONE</b>	Voto in trentesimi
<b>ORARIO DI RICEVIMENTO DEGLI STUDENTI</b>	LO PRESTI LILIANA Martedì    16:00    17:00

<p><b>PREREQUISITI</b></p>	<p>Concetto di algoritmo. Concetti di rappresentazione delle informazioni nei calcolatori. Conoscenza di semplici strutture dati. Conoscenza di base del linguaggio C.</p>
<p><b>RISULTATI DI APPRENDIMENTO ATTESI</b></p>	<p>- Conoscenza e capacita' di comprensione Lo studente al termine del corso avra' conoscenza delle problematiche inerenti la programmazione e la progettazione ad oggetti e lo sviluppo e manutenzione di software scritto in Java. In particolare lo studente conoscerà nel dettaglio caratteristiche del linguaggio Java come operatori, funzioni, strutture di controllo, classi e oggetti, ereditarietà, polimorfismo, interfacce, metodi e classi generiche, gestione delle eccezioni, collezioni, sistema di I/O. Per il raggiungimento di questo obiettivo il corso comprende: lezioni frontali; analisi e discussione di frammenti di programmi.</p> <p>- Capacita' di applicare conoscenza e comprensione Lo studente sara' in grado di progettare e sviluppare software complessi utilizzando le caratteristiche avanzate di Java; sapra' intervenire su software esistente al fine di correggerne o incrementarne le funzionalita'. Per il raggiungimento di questo obiettivo il corso comprende esercitazioni da svolgere in aula informatica comprendenti la scrittura di programmi in Java.</p> <p>- Autonomia di giudizio Lo studente sara' in grado di analizzare classi e librerie di terze parti e valutarne la possibile adozione nello sviluppo di software complessi. Per il raggiungimento di questo obiettivo il corso comprende: analisi e discussioni su casi di studio; una presentazione sommaria della libreria standard e delle principali strutture dati e discussioni su possibili vantaggi e svantaggi derivanti dal loro uso.</p> <p>- Abilita' comunicative Lo studente acquisira' la capacita' di comunicare ed esprimere problematiche inerenti l'oggetto del corso. Sara' in grado di sostenere conversazioni su tematiche di sviluppo software, di evidenziare problemi relativi a progetto e implementazione e di offrire soluzioni. Per il raggiungimento di questo obiettivo il corso comprende esercitazioni in aula informatica in cui gli studenti espongono come risolvono gli esercizi e le eventuali difficolta' incontrate.</p> <p>- Capacita' d'apprendimento Lo studente sara' in grado di affrontare in autonomia qualsiasi problematica relativa allo sviluppo di software con linguaggi ad oggetti. Sara' in grado di approfondire tematiche complesse quali il polimorfismo, la gestione dinamica della memoria, la gestione della concorrenza, la scalabilita' del software, etc... Per il raggiungimento di questo obiettivo il corso comprende: esercitazioni da svolgere autonomamente; discussione sulle eventuali difficolta' incontrate.</p>
<p><b>VALUTAZIONE DELL'APPRENDIMENTO</b></p>	<p>Il voto finale sara' dato in trentesimi e variera' da 18/30 a 30/30 con lode. Sono previste due prove: una prova pratica (in aula informatica) durante la quale sara' chiesto al candidato di scrivere un programma completo in Java a partire dalla descrizione testuale del problema. I candidati che raggiungono la sufficienza (18/30) nella prova pratica sono ammessi a sostenere la discussione orale sugli argomenti del programma. La discussione orale mirera' ad evidenziare in quale misura il candidato ha compreso ed e' in grado di padroneggiare i concetti base della programmazione ad oggetti. Il voto finale sara' calcolato come media tra i voti ottenuti nella prova pratica e nella discussione orale.</p> <p>Requisito minimo per il superamento dell'esame e' la dimostrata conoscenza, nelle due prove, delle nozioni relative a classi e oggetti, ereditarietà, polimorfismo, gestione delle eccezioni.</p> <p>La valutazione dell'apprendimento sara' focalizzata sulla valutazione dei risultati attesi (si veda sotto) in accordo con i descrittori di Dublino.</p> <p>- Valutazione della capacita' di applicare conoscenza e comprensione Per la verifica di questo obiettivo l'esame comprende la scrittura di un programma completo in Java a partire dalla descrizione testuale del problema da risolvere. L'obiettivo contribuira' per il 35% al voto finale.</p> <p>- Valutazione dell'autonomia di giudizio Per la verifica di questo obiettivo l'esame comprende la scrittura di un programma completo in Java in cui lo studente e' obbligato a effettuare delle scelte progettuali in autonomia. L'obiettivo contribuira' per il 15% al voto finale.</p> <p>- Valutazione della conoscenza e capacita' di comprensione</p>

	<p>Per la verifica di questo obiettivo l'esame comprende la discussione orale sugli argomenti del programma. L'obiettivo contribuirà per il 30% al voto finale.</p> <p>- Valutazione della capacità di apprendere</p> <p>Per la verifica di questo obiettivo l'esame comprende la discussione su alcuni argomenti introdotti a lezione e il cui approfondimento è lasciato agli studenti. L'obiettivo contribuirà per il 10% al voto finale.</p> <p>- Valutazione delle abilità comunicative</p> <p>Per la verifica di questo obiettivo l'esame comprende un esame orale sugli argomenti del corso e la discussione sul programma in Java scritto. L'obiettivo contribuirà per il 10% al voto finale.</p>
<b>OBIETTIVI FORMATIVI</b>	Il corso tratta in maniera approfondita la programmazione a oggetti in Java. Verranno trattati sia gli aspetti relativi alla programmazione ad oggetti in generale che aspetti peculiari del linguaggio Java quali l'organizzazione del codice in package, la documentazione del codice, le funzionalità per la programmazione concorrente e la libreria standard. Verranno infine trattate le strutture dati di uso più frequente inquadrando nel contesto delle librerie Java.
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	Lezioni frontali, esercitazioni in aula informatica
<b>TESTI CONSIGLIATI</b>	CLAUDIO DE SIO CESARI (2014). Manuale di Java 8. Programmazione orientata agli oggetti con Java standard edition 8. Hoepli PELLEGRINO PRINCIPE (2014). Java 8. Apogeo. PATRICK NIEMEYER. Learning Java 4th edition. O'Really

### PROGRAMMA

ORE	Lezioni
2	Introduzione a Java
2	Componenti fondamentali di un programma Java: le basi della programmazione object-oriented, i metodi in Java, le variabili, i metodi costruttori, i package.
4	Identificatori, tipi di dati e array: i tipi di dati primitivi e non primitivi, gli array in Java.
4	Operatori e gestione del flusso di esecuzione: gli operatori di base, la gestione del flusso di esecuzione, costrutti di programmazione semplici e avanzati.
6	Incapsulamento e visibilità: i paradigmi della programmazione a oggetti, l'incapsulamento, i modificatori d'accesso, il modificatore static.
4	Ereditarietà e interfacce: l'ereditarietà, il modificatore final, la classe Object, ereditarietà e incapsulamento, le interfacce.
4	Polimorfismo: il polimorfismo per metodi, overload e override, il polimorfismo per dati.
4	Eccezioni e asserzioni: il meccanismo per la gestione delle eccezioni in Java, la propagazione dell'eccezione, introduzione alle asserzioni.
2	Enumerazioni e tipi innestati: le classi interne, le classi anonime, i tipi enumerazione.
2	Tipi generici: creare propri tipi generici, la deduzione automatica del tipo.
4	Libreria: le classi String, Object, System, Runtime e Math, il package java.util, Date-Time API.
2	Gestione dei thread: la classe Thread, la sincronizzazione di thread, la comunicazione fra thread, la gestione della concorrenza.
4	Collezioni: interfacce Collection, List, Queue e Deque, Map e SortedMap, algoritmi di utilità.
2	Input/Output: character e byte stream, la lettura di input da tastiera, la gestione dei files, la serializzazione di oggetti.
2	Interfacce grafiche: JavaFX, la creazione di interfacce con i Layout, la gestione degli eventi.
ORE	Esercitazioni
6	Ambiente di sviluppo e compilazione. Semplici programmi in Java.
3	Implementazione di semplici algoritmi in Java.
3	Implementazione di programmi che utilizzano array.
6	Implementazione di semplici classi.
6	Implementazione di gerarchie di classi e polimorfismo.
3	Utilizzo di enumerazioni e tipi generici.
3	Uso della libreria di Java
3	Programmi che utilizzano collezioni e il sistema di I/O.
3	Sviluppo di prove di esami di anni precedenti.