

Principali informazioni sull'insegnamento	
Titolo insegnamento	Programmazione II
Corso di studio	Informatica e Tecnologie per la Produzione del Software
Crediti formativi	7 (Teoria) + 2 (Esercitazione/Laboratorio)
Denominazione inglese	Computer Programming II
Obbligo di frequenza	NO
Lingua di erogazione	Italiano

Docente responsabile	Nome Cognome	Indirizzo e-mail
	Pasquale Ardimento	pasquale.ardimento@uniba.it

Dettaglio credi formativi	Ambito disciplinare	SSD ING-INF/05	Crediti 7+2

Modalità di erogazione	
Periodo di erogazione	Primo semestre
Anno di corso	Secondo
Modalità di erogazione	Lezioni frontali Esercitazioni in aula Esercitazioni in laboratorio

Organizzazione della didattica	
Ore totali	225 (175 Teoria + 50 Laboratorio)
Ore di corso	86 (56 Teoria + 30 Laboratorio)
Ore di studio individuale	139 (119 Teoria + 20 Laboratorio)

Calendario	
Inizio attività didattiche	24 settembre 2018
Fine attività didattiche	11 gennaio 2019

Syllabus	
Prerequisiti	
Risultati di apprendimento previsti (declinare rispetto ai Descrittori di Dublino) (si raccomanda che siano coerenti con i risultati di apprendimento del CdS, compreso i risultati di apprendimento trasversali)	<ul style="list-style-type: none"> • <i>Conoscenza e capacità di comprensione</i> Lo studente dovrà acquisire le conoscenze di base ed alcuni aspetti avanzati della programmazione orientata agli oggetti nel linguaggio JAVA. Partendo dal paradigma di programmazione imperativo, lo studente dovrà acquisire i concetti di classe e oggetto ed i concetti di ereditarietà e polimorfismo. Lo studente dovrà, inoltre, approfondire gli aspetti relativi al trattamento delle eccezioni, all'identificazione di tipo al run-time (RTTI), alla programmazione generica, al sistema I/O • <i>Conoscenza e capacità di comprensione applicate</i> Lo studente dovrà acquisire le competenze necessarie per lo sviluppo e la realizzazione di semplici applicazioni di cui si forniscono le specifiche di massima. • <i>Autonomia di giudizio</i>

	<p>Lo studente deve dimostrare di aver acquisito una notevole autonomia di giudizio in quanto egli deve essere in grado di saper decidere in autonomia quale algoritmo applicare in base al problema da risolvere e quali scelte progettuali dover applicare per raggiungere la soluzione in modo efficace ed efficiente.</p> <ul style="list-style-type: none"> • <i>Abilità comunicative</i> <p>Lo studente deve essere in grado di illustrare in modo appropriato e talvolta dettagliato gli aspetti teorici acquisiti durante il corso e dimostrare di saperli correlare agli aspetti pratici della programmazione Object-Oriented in Java.</p> <ul style="list-style-type: none"> • <i>Capacità di apprendere</i> <p>Lo studente dovrà mostrare di aver sviluppato capacità di apprendere e di orientarsi agilmente nelle problematiche che richiedono apprendimento autonomo. Ad esempio, lo studente deve essere in grado di apprendere autonomamente il funzionamento di una libreria Java non vista a lezione, ritrovando autonomamente tutte le informazioni dettagliate di cui ha bisogno mediante la <i>Javadoc</i> disponibile online.</p>
<p>Contenuti di insegnamento</p>	<p>La programmazione orientata agli oggetti.</p> <p>Fondamenti: oggetti, classi concrete, classi astratte, metaclassi, ereditarietà singola ed ereditarietà multipla, polimorfismo, gerarchia di classi e gerarchia di interfacce. Composizione di classi. Confronto tra ereditarietà e composizione nel riuso del software. Ambienti e linguaggi di programmazione.</p> <p>Java: caratteristiche generali del linguaggio; Java ed Internet; Java vs. C++. Ambienti di sviluppo Java.</p> <p>Oggetti in Java: costruttori; distruttori; metodi, argomenti e valori di ritorno.</p> <p>Controllare il flusso di esecuzione: uso degli operatori Java; il controllo di esecuzione; l’inizializzazione.</p> <p>Nascondere le implementazioni: i package; i modificatori di accesso; le interfacce.</p> <p>Il riuso delle classi in Java: ereditarietà, derivazione protetta; polimorfismo. I contenitori: array; collezioni; le nuove collezioni.</p> <p>Trattamento delle eccezioni in Java: Sollevamento e gestione delle eccezioni; eccezioni standard e personalizzate;</p> <p>Programmazione generica in Java: <i>Generics</i> e <i>container</i>, <i>Generics</i> ed interfacce, metodi generici, il problema dell’<i>erasure</i>;</p> <p>Identificazione di tipo al run-time: RTTI (Run-Time Type Identification) “tradizionale”; il meccanismo della <i>reflection</i>;</p> <p>Il sistema I/O di Java: librerie di Input; librerie di Output; reindirizzamento e compressione dei dati;</p>

	Esercitazioni sull'ambiente di sviluppo <i>Eclipse</i> : progetto di applicazioni con più classi organizzate gerarchicamente e in package; progetto di applicazioni con classi astratte e uso del polimorfismo; progetto di applicazioni con contenitori e trattamento delle eccezioni; progetto di applicazioni con I/O da file; progetto di connessione a database;
--	---

Programma	
Testi di riferimento	<ul style="list-style-type: none"> • Bruce Eckel Thinking in Java, 4th Edition (cap. 1-11, 13-14, 16-17, 19-20, 23-24) Prentice-Hall, 2006 • Walter Savitch. Programmazione di base e avanzata con Java 2/ed Pearson Education, 2014 • Getting Started with JavaFX. https://docs.oracle.com/javase/8/javase-clienttechnologies.htm • David J. Eck: Introduction to Programming Using Java, 7th ed. (v.7.0.2) 2015/16 • J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley: The Java® Language Specification https://docs.oracle.com/javase/specs/
Note ai testi di riferimento	I testi di riferimento saranno integrati con slide e materiale didattico messo a disposizione dal docente sulla piattaforma ADA
Metodi didattici	Lezioni frontali ed esercitazioni pratiche di programmazione in Java
Metodi di valutazione	<p>Prova di laboratorio.</p> <p>La prova, della durata di 120 minuti, consiste nel realizzare un programma scritto in JAVA rispondente al problema riportato nella traccia data applicando i concetti appresi durante le lezioni.</p> <p>L'insieme di tutte le esercitazioni svolte saranno oggetto di valutazione con votazione espressa in trentesimi. Qualora la valutazione delle esercitazioni svolte da uno studente dovesse risultare positiva, lo studente sarà esonerato dal sostenere la prova di laboratorio e potrà procedere con la verbalizzazione.</p>
Criteri di valutazione	<p>Conoscenza e capacità di comprensione applicate.</p> <p>Durante lo svolgimento della prova di laboratorio lo studente dovrà dimostrare la padronanza dei costrutti, dei concetti e dei principi di base dell'Object Orientation applicandoli opportunamente per la traccia data.</p>