



Insegnamento	Programmazione ad Oggetti
Livello e corso di studio	Laurea Triennale in Ingegneria Elettronica e Informatica (L8)
Settore scientifico disciplinare (SSD)	ING-INF/05 - Sistemi di elaborazione delle informazioni
Anno di corso	3
Numero totale di crediti	9
Propedeuticità	-
Docente	Antonino Longo Minnolo <a href="https://ricerca.unicusano.it/author/antonino-longo/">https://ricerca.unicusano.it/author/antonino-longo/</a> Nickname: antonino-longo <a href="mailto:antonino.longo@unicusano.it">antonino.longo@unicusano.it</a> Orario di ricevimento: da concordare previo appuntamento con il docente
Presentazione	Il Corso di Programmazione ad Oggetti ha lo scopo di far acquisire allo studente una buona conoscenza dei principi che governano la programmazione a oggetti. Il Corso propone i concetti basilari della programmazione ad oggetti in generale, e li declina nello studio del linguaggio Java, con particolare riferimento agli aspetti peculiari di Java, quali l'organizzazione del codice in package, la documentazione del codice, le funzionalità per la programmazione concorrente e la libreria standard. Inoltre, obiettivo formativo del Corso è fornire allo studente una conoscenza nel dettaglio delle strutture dati di uso più frequente, inquadrando queste nel contesto della libreria Java. Le Attività associate al Corso sviluppano le competenze necessarie ad implementare algoritmi e strutture dati in programmi completi attraverso l'uso del linguaggio Java.
Obiettivi formativi	Il Corso di Programmazione ad Oggetti ha i seguenti obiettivi formativi: <ol style="list-style-type: none"><li>1. Illustrare i concetti di base nell'ambito della programmazione ad oggetti</li><li>2. Illustrare gli aspetti peculiari del linguaggio Java</li><li>3. Illustrare le tecniche di sviluppo di programmi secondo il paradigma della programmazione ad oggetti attraverso l'uso del linguaggio Java</li><li>4. Illustrare le strutture dati tipiche della programmazione ad oggetti</li><li>5. Orientare l'approccio, allo sviluppo, alla costruzione di algoritmi e alla strutturazione e gestione dei dati</li></ol>
Prerequisiti	La frequenza al Corso richiede la conoscenza dei concetti fondamentali di programmazione come apprese nel Corso di Sistemi di Elaborazione. Al riguardo, si consiglia di rivedere tali nozioni, propedeutiche per l'apprendimento e l'approfondimento di gran parte degli argomenti del corso.



Risultati di  
apprendimento attesi

## **Conoscenza e capacità di comprensione (KNOWLEDGE AND UNDERSTANDING)**

Lo studente al termine del Corso avrà conoscenza delle problematiche inerenti la programmazione e la progettazione ad oggetti e lo sviluppo e manutenzione di software scritto in Java. Inoltre, lo studente acquisirà la conoscenza nel dettaglio delle 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. Inoltre, tramite le Etivity gli studenti acquisiranno la capacità di impiegare i principali ambienti di programmazione Java.

## **Applicazione delle conoscenze (APPLYING KNOWLEDGE AND UNDERSTANDING)**

Lo studente sarà in grado di progettare e sviluppare software complessi utilizzando le caratteristiche avanzate di Java; sarà inoltre in grado di intervenire su software esistente al fine di correggerne o incrementarne le funzionalità.

Le Etivity prevedono l'applicazione delle conoscenze teoriche all'implementazione di semplici programmi in linguaggio Java, utilizzando gli appositi strumenti e ambienti di sviluppo.

## **Capacità di trarre conclusioni (ABILITY TO DRAW CONCLUSIONS)**

Lo studente sarà in grado di individuare classi e librerie di terze parti più appropriate per i servizi da sviluppare, analizzandone l'aderenza ai requisiti che ne derivano, valutandone dunque la possibile adozione nello sviluppo di software complessi.

## **Abilità comunicative (COMMUNICATION SKILLS)**

Lo studente sarà in grado di descrivere e sostenere conversazioni su tematiche relative allo sviluppo di software basato sulla programmazione ad oggetti, di evidenziare problemi relativi a progetto e implementazione, e di offrire soluzioni, adoperando una terminologia adeguata.

## **Capacità di apprendere (LEARNING SKILLS)**

Lo studente al termine del Corso avrà conoscenza delle nozioni fondamentali necessarie per la progettazione e lo sviluppo di applicazioni e servizi secondo il paradigma della programmazione ad oggetti. Questo gli consentirà di affrontare in autonomia qualsiasi problematica relativa allo sviluppo di software con linguaggi ad oggetti, essendo in grado di approfondire tematiche complesse quali il polimorfismo, la gestione dinamica della memoria, la gestione della concorrenza, e la scalabilità del software, fornendo basi utili per poter apprendere quanto verrà proposto nei corsi specialistici di ingegneria informatica, con particolare riferimento gli argomenti legati alla conoscenza dei linguaggi e dei paradigmi di programmazione.

Organizzazione  
dell'insegnamento

Il corso è sviluppato attraverso le **lezioni preregistrate audio-video** che compongono, insieme a slide e dispense, i materiali di studio disponibili in piattaforma.



Sono poi proposti dei **test di autovalutazione**, di tipo asincrono, che corredano le lezioni preregistrate e consentono agli studenti di accertare sia la comprensione, sia il grado di conoscenza acquisita dei contenuti di ognuna delle lezioni.

La **didattica interattiva** è svolta nel forum della “classe virtuale” e comprende 8 **Etivity**. In particolare, il Corso di Ingegneria del Software prevede 9 Crediti formativi. Il carico totale di studio per questo modulo di insegnamento è compreso tra 220 e 250 ore così suddivise in:

- **circa 192,5 ore** per la visualizzazione e lo studio del materiale videoregistrato (28 Ore videoregistrate di Teoria e 7,5 ore di esercitazioni);
- **circa 40 ore di Didattica Interattiva** per l’elaborazione e la consegna di 8 Etivity;
- **circa 5 ore di Didattica Interattiva** per l’esecuzione dei test di autovalutazione.

Si consiglia di distribuire lo studio della materia uniformemente in un periodo di 9 settimane dedicando tra le 20 e le 30 ore di studio a settimana

## Contenuti del corso

**Modulo 1 – Introduzione alla programmazione ad oggetti e Java (11 lezioni di teoria videoregistrate per un impegno di 42 ore – settimana 1 e 2)** dove sono affrontati i seguenti argomenti: Introduzione a Java. Componenti fondamentali di un programma Java: le basi della programmazione object-oriented, i metodi in Java, le variabili, i metodi costruttori, i package. Esercitazione su ambiente di sviluppo e compilazione di semplici programmi in Java.

**Etivity 1 (8 ore di carico di studio – settimana 1) – Ambiente di sviluppo e compilazione. Semplici programmi in Java.**

**Modulo 2 – Tipi di dato (9 lezioni di teoria videoregistrate per un impegno di 35 ore – settimana 2 e 3)** dove sono affrontati i seguenti argomenti: Identificatori, tipi di dati e array: i tipi di dati primitivi e non primitivi, gli array in Java. Esercitazione sugli array.

**Etivity 2 (4 ore di carico di studio – settimana 2) – Implementazione di programmi che usano array.**

**Modulo 3 – Gestione flusso di esecuzione (5 lezioni di teoria videoregistrate per un impegno di 17,5 ore – settimana 2 e 3)** dove sono affrontati i seguenti argomenti: Operatori e gestione del flusso di esecuzione: gli operatori di base, la gestione del flusso di esecuzione, costrutti di programmazione semplici e avanzati. Esercitazione sull’uso dei costrutti.

**Etivity 3 (4 ore di carico di studio – settimana 3) – Implementazione di semplici algoritmi in Java.**



**Modulo 4 – Incapsulamento (4 lezioni di teoria videoregistrate per un impegno di 14,5 ore e 1 lezione di esercitazione per un impegno di 2,5 ore – settimana 4)** dove sono affrontati i seguenti argomenti: Incapsulamento e visibilità: i paradigmi della programmazione a oggetti, l'incapsulamento, i modificatori d'accesso, il modificatore static. Esercitazione su incapsulamento e visibilità.

**Etivity 4 (8 ore di carico di studio – settimana 3-4) – Implementazione di semplici classi.**

**Modulo 5 – Ereditarietà e polimorfismo (6 lezioni di teoria videoregistrate per un impegno di 21 ore – settimana 4 e 5)** dove sono affrontati i seguenti argomenti: Ereditarietà e interfacce: l'ereditarietà, il modificatore final, la classe Object, ereditarietà e incapsulamento, le interfacce; Polimorfismo: il polimorfismo per metodi, overload e override, il polimorfismo per dati. Esercitazione su ereditarietà e polimorfismo.

**Etivity 5 (8 ore di carico di studio – settimana 5) – Implementazione di programmi che fanno uso di gerarchie di classi e polimorfismo.**

**Modulo 6 – Eccezioni e tipi generici (4 lezioni di teoria videoregistrate per un impegno di 13,5 ore – settimana 6)** dove sono affrontati i seguenti argomenti: Eccezioni e asserzioni: il meccanismo per la gestione delle eccezioni in Java, la propagazione dell'eccezione, introduzione alle asserzioni; Enumerazioni e tipi innestati: le classi interne, le classi anonime, i tipi enumerazione; Tipi generici: creare propri tipi generici, la deduzione automatica del tipo. Esercitazione sull'uso di eccezioni e tipi generici.

**Etivity 6 (4 ore di carico di studio – settimana 6) – Implementazione di programmi che fanno uso di eccezioni e tipi generici.**

**Modulo 7 – Librerie e concorrenza (7 lezioni di teoria videoregistrate per un impegno di 24,5 ore – settimana 6 e 7)** dove sono affrontati i seguenti argomenti: Libreria: le classi String, Object, System, Runtime e Math, il package java.util, Date-Time API; Gestione dei thread: la classe Thread, la sincronizzazione di thread, la comunicazione fra thread, la gestione della concorrenza. Esercitazione sull'uso della libreria di Java e sulla gestione della concorrenza.

**Etivity 7 (4 ore di carico di studio – settimana 7) – Implementazione di programmi che fanno uso della libreria di Java e dei thread.**

**Modulo 8 – Collezioni, I/O e GUI (6 lezioni di teoria videoregistrate per un impegno di 24,5 ore – settimana 8)** dove sono affrontati i seguenti argomenti: Collezioni: interfacce Collection, List, Queue e Deque, Map e SortedMap, algoritmi di utilità; Input/Output:



	<p>character e byte stream, la lettura di input da tastiera, la gestione dei files, la serializzazione di oggetti; Interfacce grafiche: JavaFX, la creazione di interfacce con i Layout, la gestione degli eventi. Esercitazione sull'uso di collezioni, del sistema di I/O, di interfacce grafiche.</p> <p><b>Etivity 8 (8 ore di carico di studio – settimana 8) – Implementazione di programmi che fanno uso di collezioni, I/O, GUI</b></p> <p><b>Etivity 9 (6 ore di carico di studio – settimana 8) – Simulazione esame.</b></p> <p><b>Esercitazioni su compiti d'esame (2 lezioni di esercitazione per un impegno di 5 ore)</b></p>
Materiali di studio	<p>MATERIALI DIDATTICI A CURA DEL DOCENTE</p> <p>Il materiale didattico presente in piattaforma è suddiviso in 8 moduli. Essi ricoprono interamente il programma e ciascuno di essi contiene dispense, slide e videolezioni in cui il docente commenta le slide. Tale materiale contiene tutti gli elementi necessari per affrontare lo studio della materia.</p> <p>Testi consigliati:</p> <ul style="list-style-type: none"><li>• Herbert Schildt, "Java la guida completa". McGrawill – ISBN 9788838667664</li></ul>
Modalità di verifica dell'apprendimento	<p>L'esame consiste nello svolgimento di una prova scritta tendente ad accertare le capacità di analisi e rielaborazione dei concetti acquisiti.</p> <p>In accordo con il modello formativo del Corso di Studi, La valutazione finale dell'insegnamento, espressa in trentesimi, prende in considerazione anche l'attività svolta in itinere dallo studente e valutata attraverso il punteggio assegnato alle otto Etivity proposte (da 0 a 6).</p> <p>La prova scritta se svolta per intero (si veda più avanti) prevede, di norma, due esercizi per accertare le capacità di analisi e rielaborazione dei concetti acquisiti ed una domanda di teoria.</p> <p>I risultati di apprendimento attesi circa le conoscenze della materia e la capacità di applicarle sono valutate dalla prova scritta, mentre le abilità comunicative, la capacità di trarre conclusioni e la capacità di autoapprendimento sono valutate in itinere attraverso le Etivity.</p> <p>È possibile effettuare l'esame attraverso due prove parziali scritte</p> <ul style="list-style-type: none"><li>✓ Parte - 1 moduli da 1 a 4;</li><li>✓ Parte - 2 moduli da 5 a 8.</li></ul> <p>In sede di valutazione finale, si terrà conto anche della proficua partecipazione ai forum (aule virtuali) e al corretto svolgimento delle e-tivity proposte.</p>
Criteri per l'assegnazione dell'elaborato finale	<p>L'assegnazione dell'elaborato finale avverrà sulla base di un colloquio con il docente in cui lo studente manifesterà i propri specifici interessi in relazione a qualche argomento che</p>



# CUNIVERSITÀ CUSANO

intende approfondire; non esistono preclusioni alla richiesta di assegnazione della tesi e non è prevista una media particolare per poterla richiedere.