

<b>Denominazione</b>	PROGRAMMAZIONE OBJECT ORIENTED
<b>Moduli componenti</b>	-
<b>Settore scientifico-disciplinare</b>	IINF-05/A (ex ING-INF/05)
<b>Anno di corso e semestre di erogazione</b>	2° anno, 1° semestre
<b>Lingua di insegnamento</b>	Italiano
<b>Carico didattico in crediti formativi universitari</b>	6
<b>Numero di ore di attività didattica assistita complessive e ripartite tra DE e DI</b>	48 (pari a 6 CFU di Didattica Erogativa) + 6 ore di Didattica Interattiva
<b>Docente</b>	Ing. Giovanni Maria Biancofiore
<b>Risultati di apprendimento specifici</b>	<p>Il corso mira a fornire agli studenti una solida comprensione dei concetti di base della programmazione orientata agli oggetti (OO) e del relativo paradigma attraverso l'uso di linguaggi object-oriented come Python.</p> <p><i>Conoscenze e comprensione.</i> Lo studente acquisirà le conoscenze relative a: modellazione delle informazioni seguendo il paradigma OO; metodologie e tecniche avanzate di programmazione; linguaggi di programmazione orientati agli oggetti (Python).</p> <p><i>Capacità di applicare conoscenze e comprensione.</i> Lo studente svilupperà competenze per essere in grado di: studiare e valutare autonomamente un problema e le sue possibili soluzioni; ideare e formalizzare una soluzione algoritmica sfruttando il paradigma OO.</p> <p><i>Autonomia di giudizio e pensiero critico:</i> Al termine del corso lo studente sarà in grado di gestire problematiche relative alla risoluzione di problemi che richiedono l'implementazione della soluzione individuata tramite linguaggi di programmazione orientati agli oggetti. Saprà inoltre comunicare con chiarezza le metodologie utilizzate e i risultati ottenuti.</p> <p><i>Abilità comunicative:</i> Al termine del corso lo studente sarà in grado di presentare e spiegare con chiarezza e precisione le implementazioni proposte; documentare il codice e le soluzioni in modo comprensibile per altri programmatori e stakeholder tecnici; partecipare attivamente a discussioni e lavori di gruppo, esponendo idee e soluzioni in modo efficace; utilizzare un linguaggio tecnico appropriato nelle comunicazioni scritte e orali.</p> <p><i>Capacità di apprendere:</i> Lo studente svilupperà competenze di autoapprendimento per adattarsi a nuovi linguaggi di programmazione e paradigmi computazionali; saprà utilizzare le risorse didattiche e la documentazione per risolvere problemi complessi in modo autonomo. Inoltre, sarà in grado di apprendere nuove tecnologie e strumenti attraverso lo studio indipendente e la pratica continua.</p>
<b>Programma</b>	<p>Il programma del corso è composto dai seguenti contenuti didattici:</p> <ul style="list-style-type: none"> <li>● Teoria della programmazione orientata agli oggetti <ul style="list-style-type: none"> <li>○ Principi e caratteristiche OOP</li> <li>○ Classi, oggetti, attributi e metodi</li> <li>○ Diagramma delle classi UML</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ Costruttori e Distruttori</li> <li>○ Polimorfismo, Overloading vs Overriding, Operator Overloading</li> <li>○ Incapsulamento, Variabili, locali, non locali e globali</li> <li>○ Sottoclassi, Ereditarietà e Gerarchie, Composizione, ereditarietà multipla</li> <li>○ Interfacce, Abstract Base Classes</li> <li>○ Implementation Inheritance vs Interface Inheritance</li> </ul> <ul style="list-style-type: none"> <li>● Programmazione OO in Python e costrutti specifici del linguaggio</li> </ul>
<b>Tipologie di attività didattiche previste e relative modalità di svolgimento</b>	<p>L'insegnamento è strutturato in lezioni di didattica frontale, incoraggiando l'interazione e la partecipazione attiva degli studenti, ed esercitazioni integrate con le lezioni e svolte con l'ausilio di un elaboratore. Saranno mostrati esempi di progettazione ed implementazione di applicativi basati su programmazione orientata agli oggetti.</p> <p>È previsto inoltre l'utilizzo di tecnologie digitali per l'erogazione delle lezioni e delle esercitazioni.</p>
<b>Metodi e criteri di valutazione dell'apprendimento</b>	<p>La valutazione dell'apprendimento (sia per studenti frequentanti che non frequentanti) consiste nello svolgimento di una prova scritta contenente quesiti teorici e pratici relativi ai contenuti del corso proposti attraverso domande a risposta multipla. Sarà richiesta inoltre la definizione e codifica di un programma in linguaggio Python. La valutazione del quesito progettuale terrà conto della comprensione delle specifiche di progetto e della qualità implementativa della soluzione proposta.</p> <p>Sarà prevista inoltre una prova orale facoltativa. La valutazione del colloquio orale è espressa in trentesimi e terrà conto della proprietà di linguaggio, della capacità argomentativa, di analisi critica e di ragionamento.</p>
<b>Criteri di misurazione dell'apprendimento e di attribuzione del voto finale</b>	<p>La valutazione dell'apprendimento prevede l'attribuzione di un voto finale espresso in trentesimi.</p> <p>Il voto finale sarà determinato attraverso l'esame scritto sopra dettagliato. La lode sarà valutata per i soli studenti che abbiano raggiunto la valutazione complessiva di 30/30 analizzando la capacità di proporre soluzioni corrette ed efficienti nella risoluzione di problemi.</p>
<b>Propedeuticità</b>	Fondamenti di Informatica
<b>Materiale didattico utilizzato e materiale didattico consigliato</b>	<ul style="list-style-type: none"> <li>- Slides fornite dal docente</li> <li>- Concetti di informatica e fondamenti di Python. Cay Horstmann - Rance D. Necaie. ISBN 8891635433. Ean 9788891635433. Apogeo Education. Maggioli Editore.</li> </ul>