



Principali informazioni sull'insegnamento	
Denominazione dell'insegnamento	Informatica
Corso di studio	Fisica (L-30)
Anno di corso	I
Crediti formativi universitari (CFU)	8
SSD	ING-INF/05
Lingua di erogazione	Italiano
Periodo di erogazione	II Semestre (Marzo - Giugno)
Obbligo di frequenza	NO (Fortemente consigliata)

Docente	
Nome e cognome	Teresa Maria Altomare Basile
Indirizzo mail	teresamaria.basile@uniba.it
Telefono	080 544 2235
Sede	Dipartimento Interateneo di Fisica M. Merlin - stanza 235
Sede virtuale (Codice Microsoft Teams)	/
Ricevimento (giorni, orari e modalità)	Martedì ore 10:30 - 12:30 Stanza 235 - Il Piano - Dipartimento di Fisica su appuntamento - in presenza o in modalità telematica (piattaforma MSTeams)- da concordare con il docente via email

Syllabus	
Obiettivi formativi	<i>Acquisire i fondamenti concettuali del problem solving, dell'architettura e della programmazione dei calcolatori. Acquisire la capacità di progettare e realizzare applicazioni per la risoluzione di problemi in campo scientifico e in ambito più generale.</i>
Prerequisiti	<i>Non vi sono prerequisiti specifici differenti da quelli richiesti per l'accesso al corso di laurea.</i>
Contenuti di insegnamento (Programma)	<p>1. <i>Nozioni basilari di Informatica. Elaborazione dei dati - L'architettura di un calcolatore: la macchina di Von Neumann. Codifica di dati e istruzioni di programma. Esecuzione di programmi: il ciclo fetch-decode-execute. Istruzioni aritmetico/logiche. L'informazione e la sua codifica - Bit e loro memorizzazione. Rappresentazione dell'informazione come configurazione di bit. Sistema binario. Rappresentazione di interi e razionali.</i></p> <p>2. <i>Problem Solving - Dal problema all'algoritmo. Tecniche di decomposizione di problemi: Sequenziale, Selettiva, Iterativa, Ricorsiva. Programmazione Strutturata. Rappresentazione degli algoritmi: Diagrammi di flusso e Linguaggio lineare</i></p> <p>3. <i>Principi di programmazione imperativa - dall'algoritmo al programma. Dati e tipi di dati, istruzioni primitive e strutture di controllo (sequenza, selezione, iterazione). Sottoprogrammi e Astrazione Funzionale. Visibilità di sottoprogrammi e variabili. Parametri formali ed effettivi - Passaggio di parametri. Record di Attivazione e Ricorsione. Linguaggi di Programmazione.</i></p> <p>4. <i>Introduzione al linguaggio - Python Sintassi del linguaggio. Gestione dell'I/O. Dati e istruzioni. Controllo del flusso di esecuzione condizionale: Istruzioni if, if/else e elif. Controllo del flusso mediante iterazioni: costrutti while e for. Cicli definiti vs cicli indefiniti. Cicli annidati. Definizione/uso di funzioni e librerie esterne. Funzioni ricorsive. Strutture di dati dinamiche: liste, insiemi, dizionari. Sintassi, proprietà basilari e operazioni. File stream: manipolazione di file di testo e file csv. Cenni su principi di programmazione orientata ad oggetti. Classi e oggetti in Python e loro utilizzo nella definizione di tipi di dati e strutture di dati.</i></p>

	<p>5. Algoritmi. Algoritmi elementari: calcolo del minimo/massimo fra n numeri, scambio, conteggio, sommatoria di un insieme di numeri, inversione cifre di un intero positivo. Algoritmi elementari su sequenze lineari di dati (array): ricerca del massimo/minimo, calcolo di grandezze statistiche (media, dev. standard, moda, mediana), costruzione e visualizzazione di istogrammi, palindroma, inversione degli elementi e rimozione valori duplicati. Algoritmi di ordinamento, ricerca e fusione su sequenze lineari di dati: ordinamento per selezione, per inserzione, a bolle (bubblesort), per fusione (mergesort) ricerca lineare e binaria, fusione di 2 array ordinati, partizionamento rispetto ad un valore. Algoritmi su matrici: somma e prodotto di matrici, trasposta, verifiche proprietà su matrici. Algoritmi iterativi vs ricorsivi: fattoriale, serie di Fibonacci, massimo comun divisore, ricerca binaria.</p> <p>6. Esercitazioni in laboratorio. Sviluppo di applicazioni che implementano gli algoritmi elementari.</p> <p>Nota: una versione dettagliata degli esercizi proposti e svolti durante le esercitazioni di laboratorio è messa a disposizione sulla pagina di riferimento dell'insegnamento durante il periodo di svolgimento delle lezioni.</p>
Testi di riferimento	<p>J. Glenn Brookshear, Dennis Brylow. <i>Informatica. Una panoramica generale.</i> Pearson Cay S. Horstmann, Rance D. Necaise. <i>Concetti di informatica e fondamenti di Python.</i> Apogeo Tony Gaddis. <i>Introduzione a Python.</i> Pearson</p>
Note ai testi di riferimento	<p>In aggiunta ai testi di riferimento, si può consultare il materiale didattico (slides) utilizzato durante le lezioni e le esercitazioni di laboratorio con la presentazione di esercizi svolti.</p>

Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (laboratorio, campo, esercitazione, altro)	Studio individuale
200	40	45	115
CFU/ETCS			
8	5	3	

Metodi didattici
<p>Lezioni in aula, supportate da slides, atte a fornire le basi, teoriche e pratiche, dell'approccio al problem solving, della programmazione imperativa (con cenni a quella orientata agli oggetti), utilizzando come linguaggio di riferimento il linguaggio di programmazione Python. Esercitazioni guidate di laboratorio, svolte con l'ausilio di pc, mirate allo sviluppo di semplici applicazioni software attraverso fasi di analisi del problema, definizione della strategiaolutiva, codifica e test del programma nel linguaggio di riferimento.</p>

Risultati di apprendimento previsti	
<p>Conoscenza e capacità di comprensione</p>	<p>0 Conoscenza della struttura del calcolatore, della codifica dell'informazione, degli elementi di programmazione e del linguaggio di riferimento Python.</p>
<p>Conoscenza e capacità di comprensione applicate</p>	<p>0 Capacità di analisi del problema, individuazione di una strategiaolutiva e sviluppo di applicazioni nel linguaggio di riferimento.</p>

Competenze trasversali	<ul style="list-style-type: none"> • Autonomia di giudizio <ul style="list-style-type: none"> ○ Capacità di individuare un metodo risolutivo ad un problema, valutando la correttezza della risoluzione e l'efficienza della scelta progettuale con le conoscenze teoriche e pratiche acquisite. • Abilità comunicative <ul style="list-style-type: none"> ○ Acquisizione del linguaggio e del formalismo dell'informatica di base, necessario per la consultazione e comprensione dei testi, per l'esposizione delle conoscenze acquisite, l'analisi e la risoluzione dei problemi. • Capacità di apprendere in modo autonomo <ul style="list-style-type: none"> ○ Capacità di consultare e comprendere materiale relativo a diversi linguaggi di programmazione, a parità di paradigma di programmazione, ed essere in grado di utilizzarlo per lo sviluppo.
------------------------	--

Valutazione	
Modalità di verifica dell'apprendimento	<p>La verifica dell'apprendimento avviene attraverso una prova di laboratorio e una prova orale. Il superamento della prova di laboratorio è propedeutico al sostenimento della prova orale. Entrambe le prove (di laboratorio e orale) devono essere sostenute nell'ambito del medesimo appello.</p> <p>La prova di laboratorio consiste nella implementazione di applicazioni per la soluzione di problemi individuando gli algoritmi e sviluppando il relativo programma in Python.</p> <p>La prova orale è relativa all'intero programma. Inizia con la discussione dello svolgimento della prova di laboratorio, seguita dalla discussione di argomenti teorici ed esempi di applicazioni.</p>
Criteri di valutazione	<ul style="list-style-type: none"> • Conoscenza e capacità di comprensione <ul style="list-style-type: none"> ○ Conoscenza dei concetti di base della programmazione: dalla rappresentazione e gestione dell'informazione agli algoritmi fondamentali. • Conoscenza e capacità di comprensione applicate <ul style="list-style-type: none"> ○ Capacità di applicare le metodologie apprese durante il corso alla risoluzione di semplici problemi applicativi. • Autonomia di giudizio <ul style="list-style-type: none"> ○ Capacità di individuare ed applicare gli strumenti più idonei, in termini di correttezza ed efficienza, nella strategia proposta per la risoluzione degli esercizi. • Abilità comunicative <ul style="list-style-type: none"> ○ Chiarezza espositiva dei concetti teorici e pratici con padronanza del linguaggio specifico dell'insegnamento. • Capacità di apprendere <ul style="list-style-type: none"> ○ Capacità di saper identificare il contesto di ogni concetto ed applicare le competenze teoriche e pratiche della programmazione in domini applicativi differenziati.
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	<p>Il voto finale è attribuito in trentesimi. L'esame si intende superato quando il voto di entrambe le prove è maggiore o uguale a 18. Alla determinazione del voto finale contribuiscono sia la prova di laboratorio (50%) che la prova orale (50%).</p>
Altro	