

Principali informazioni sull'insegnamento	
Denominazione dell'insegnamento	Informatica
Corso di studio	<i>Matematica</i>
Anno di corso	<i>2021/2022</i>
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS):	6
SSD	<i>INF/01</i>
Lingua di erogazione	<i>Italiano</i>
Periodo di erogazione	<i>1° Semestre, 1° Anno (27 settembre 2021 – 14 gennaio 2022)</i>
Obbligo di frequenza	<i>no</i>

Docente	
Nome e cognome	<i>Marco Polignano</i>
Indirizzo mail	<i>marco.polignano@uniba.it</i>
Telefono	<i>+39 080 544 2312</i>
Sede	<i>Ufficio 759 Dipartimento di Informatica, Campus Universitario, Via E. Orabona 4, 70126 Bari</i>
Sede virtuale	<i>Microsoft Teams (Marco Polignano – marco.polignano@uniba.it)</i>
Ricevimento (giorni, orari e modalità)	<i>Lenedì 16.30 -18.30 Piattaforma Microsoft Teams (per una organizzazione ottimale, si prega di prendere appuntamento via mail)</i>

Syllabus	
Obiettivi formativi	<i>Acquisire concetti base riguardanti le architetture dei calcolatori. Acquisire concetti base riguardanti metodi e tecniche di programmazione. Applicare questi concetti per risolvere problemi per mezzo del linguaggio Python. Nozioni e concetti base riguardanti algoritmi e complessità computazionale.</i>
Prerequisiti	<i>Nozioni elementari di matematica fornite dalle scuole secondarie superiori.</i>
Contenuti di insegnamento (Programma)	<i>Introduzione al corso Introduzione all'architettura dei calcolatori ed alle reti Introduzione alla programmazione Diagrammi di flusso Linguaggi di programmazione Programmazione strutturata Variabili e funzioni Strutture dati Costrutti di programmazione Introduzione alla computabilità Complessità computazionale Macchine di Turing Funzioni ricorsive Introduzione al linguaggio Python</i>
Testi di riferimento	<i>Andrew J. Tanenbaum, Architettura dei calcolatori. Un approccio strutturale. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Introduzione agli algoritmi e strutture dati, McGraw-Hill Education Slide ed altro materiale mostrato a lezione.</i>
Note ai testi di riferimento	<i>Libri consigliati ma non obbligatori. Il docente fornirà agli studenti del materiale integrativo, come articoli scientifici.</i>

Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (laboratorio, campo, esercitazione, altro)	Studio individuale
150	32	28	90
CFU/ETCS			
6	4	2	

Metodi didattici	<i>Insegnamento frontale in classe ed esercitazioni guidate. Si applicherà la modalità e-learning solo qualora richiesto dal Senato Accademico come misura da adottare per fronteggiare l'emergenza COVID-19.</i>

Risultati di apprendimento previsti	<i>Al termine del corso, lo studente sarà in grado di:</i>
Conoscenza e capacità di comprensione	<ul style="list-style-type: none"> ○ Riconoscere componenti dell'architettura di un calcolatore elettronico. ○ Descrivere formalmente un linguaggio di programmazione. ○ Conoscere le basi della programmazione. ○ Conoscere le basi dell'informatica.
Conoscenza e capacità di comprensione applicate	<ul style="list-style-type: none"> ○ Riconoscere ed usare le strutture dati e i tipi di dati primitivi. ○ Utilizzare i costrutti base di programmazione. ○ Usare correttamente ambienti e strumenti per la risoluzione di problemi complessi attraverso l'ausilio di un programma software.
Competenze trasversali	<ul style="list-style-type: none"> • Autonomia di giudizio <ul style="list-style-type: none"> ○ Decidere quali strutture dati e costrutti di programmazione utilizzare per la risoluzione di problemi complessi. ○ Identificare errori nel codice del programma software e correggerli in autonomia. ○ Scegliere software e linguaggi di programmazione adeguati alla risoluzione di problemi complessi. • Abilità comunicative <ul style="list-style-type: none"> ○ Utilizzare correttamente la terminologia Informatica. • Capacità di apprendere in modo autonomo <ul style="list-style-type: none"> ○ Risolvere in autonomia problemi complessi che richiedono l'ausilio di un programma software. ○ Comprendere e risolvere errori nel codice software.

Valutazione	
Modalità di verifica dell'apprendimento	<i>Sessioni d'esame (al termine dell'insegnamento). Esame con esercizi scritti e domande teoriche a risposta aperta. La prova avrà durata di 1.30h.</i>
Criteri di valutazione	La prova permetterà di valutare lo studente rispetto alle seguenti capacità apprese durante il corso. Conoscenza e capacità di comprensione:

	<p><i>Capacità di apprendere concetti fondamentali riguardanti metodi e tecniche di programmazione e concetti generici dell'Informatica.</i></p> <p>Conoscenza e capacità di comprensione applicate: <i>Abilità di applicare i concetti appresi per risolvere ed implementare le soluzioni a problemi di varia natura.</i></p> <p>Autonomia di giudizio: <i>Capacità di giudicare la consistenza della struttura logica usata per la costruzione degli algoritmi. Capacità di identificare gli strumenti appropriati alla risoluzione di problemi di programmazione.</i></p> <p>Abilità comunicative: <i>Apprendere i metodi ed i linguaggi dell'informatica per comprendere testi, comunicare la conoscenza appresa, descrivere, analizzare e risolvere i problemi algoritmici.</i></p> <p>Capacità di apprendere: <i>Acquisire dei metodi di apprendimento duraturi, supportati da ricerche contestuali e risolvendo esercizi e domande fatte periodicamente durante il corso.</i></p>
<p>Criteria di misurazione dell'apprendimento e di attribuzione del voto finale</p>	<p><i>Il voto finale è attribuito in trentesimi. L'esame si intende superato quando il voto è maggiore o uguale a 18. Si attribuirà un punteggio per singola domanda fino ad arrivare ad un totale di 32 punti. Si otterrà la lode con punteggio superiore a 30.</i></p>
<p>Altro</p>	<p><i>Si consiglia la frequenza, in particolare la partecipazione alle esercitazioni pratiche, che prevedono anche lo svolgimento in aula di esercitazioni. Per gli studenti non frequentanti si suggerisce di consultare il docente nelle ore di ricevimento per verificare la correttezza delle soluzioni proposte.</i></p>