

INFORMATICA (MT000004)

1. lingua insegnamento

Italiano.

2. contenuti

Coordinatore: Prof. FRATI FABRIZIO

Anno di corso: I anno

Semestre: 2° semestre

CFU: 6

Moduli e docenti incaricati:

- INFORMATICA (MT000004) - 6 CFU - SSD ING-INF/05 - Prof. Fabrizio Frati

3. testi di riferimento

Testo di riferimento:

“Programmazione in Python” di Kenneth A. Lambert, Apogeo Education, edito da Maggioli Editore.

Ulteriori testi per consultazione:

“Python - Introduzione alla programmazione” di Cay Horstmann, Rance D. Necaise, Apogeo Education, edito da Maggioli Editore.

“Pensare in Python - Come pensare da Informatico” di Allen Downey, edito da Egea.

“Python e Machine Learning” di Alessandro Bellini, Andrea Guidi, edito da McGraw-Hill.

4. obiettivi formativi

Conoscenza e capacità di comprensione (Dublino 1)

Lo studente acquisirà consapevolezza del processo informatico per la risoluzione automatica di problemi. Obiettivi specifici sono:

la conoscenza e la comprensione del funzionamento del calcolatore e della rappresentazione dell'informazione

la conoscenza e la comprensione di tecniche di modellazione informatica dei problemi

la conoscenza e la comprensione di metodologie per la progettazione di algoritmi

la conoscenza e la comprensione del processo di trasformazione di un algoritmo in un programma

la conoscenza e la comprensione di programmi e librerie per gestire e analizzare dati.

Conoscenza e capacità di comprensione applicate (Dublino 2)

Lo studente sarà in grado di applicare metodologie e strumenti tecnologici per la risoluzione automatica di problemi. Obiettivi specifici sono:

essere in grado di effettuare una modellazione informatica di problemi
essere in grado di progettare algoritmi per la risoluzione automatica di problemi
essere in grado di realizzare programmi che implementano algoritmi per la risoluzione automatica dei problemi
essere in grado di utilizzare funzioni di libreria per la gestione e l'analisi di dati.

Autonomia di giudizio (Dublino 3)

Lo studente sarà in grado di analizzare problemi di interesse, decidere una modellazione informatica appropriata per gli stessi, progettare la strategia risolutiva migliore per risolverli e quindi decidere quali funzioni, moduli e librerie realizzare ed utilizzare per risolverli.

Abilità comunicative (Dublino 4)

Lo studente sarà in grado di sintetizzare e comunicare in modo chiaro le proprie scelte di modellazione, progettazione algoritmica e implementazione.

Capacità di apprendere (Dublino 5)

Lo studente sarà in grado di integrare, attraverso lo studio autonomo, quanto appreso nel corso. Aspetti di interesse per un apprendimento autonomo includono ulteriori tecniche algoritmiche, ulteriori elementi sintattici e semantici del linguaggio di programmazione ed ulteriori funzioni di libreria.

5. prerequisiti

È richiesta la conoscenza di nozioni scolastiche di base di matematica, come nei programmi di scuola secondaria superiore.

6. metodi didattici

La maggior parte delle ore di didattica del corso sarà utilizzata per lezioni frontali, durante la quale verranno spiegati in dettaglio gli argomenti che costituiscono il programma del corso, con il supporto di slides (*Dublino 1: Conoscenza e capacità di comprensione*). Le lezioni saranno interattive e coinvolgeranno gli studenti nella formazione delle loro idee e nell'espressione dei loro dubbi sui concetti introdotti (*Dublino 4: Abilità comunicative*). Le lezioni frontali forniranno delle conoscenze che permetteranno allo studente di analizzare e modellare i problemi affrontati decidendo autonomamente la migliore strategia algoritmica per la soluzione e per la implementazione della soluzione stessa (*Dublino 3: Autonomia di giudizio*). Gli studenti potranno inoltre espandere quanto appreso nelle lezioni frontali attraverso lo studio autonomo su testi e risorse online consigliate (*Dublino 5: Capacità di apprendere*).

Una parte consistente delle ore di didattica del corso sarà utilizzata per esercitazioni in laboratorio, durante le quali gli studenti potranno fronteggiare problemi che risolveranno mettendo in atto le metodologie apprese a lezione (*Dublino 2: Conoscenza e capacità di comprensione applicate* e *Dublino 3: Autonomia di giudizio*). Gli studenti potranno interagire con il docente esponendo i loro dubbi (*Dublino 4: Abilità comunicative*) e ricevere indicazioni su come procedere verso la soluzione del problema.

7. altre informazioni

Ricevimento: ogni venerdì dalle 09:30 alle 11:30, previa richiesta via e-mail.

8. modalità di verifica dell'apprendimento

La verifica del profitto è effettuata attraverso una prova che si svolge al computer, in laboratorio, e che consiste di due parti.

La prima parte è volta a valutare la conoscenza, la capacità di comprensione e le abilità comunicative acquisite dallo studente, relativamente agli argomenti teorici del corso, come il funzionamento del calcolatore, la rappresentazione binaria dell'informazione e l'algebra Booleana. Questa parte della verifica di profitto consiste di 10 domande a risposta multipla o a risposta libera breve ed assegna un voto massimo di 10 punti.

La seconda parte è volta a valutare la conoscenza, la capacità di comprensione, l'autonomia di giudizio e la capacità di comprensione applicata relativamente agli argomenti pratici del corso, come la progettazione di algoritmi e la realizzazione di programmi. Questa parte della verifica di profitto consiste di alcuni problemi che lo studente deve risolvere modellando gli stessi, progettando un relativo algoritmo risolutivo e realizzando quindi quest'ultimo come programma. Questa parte della verifica di profitto consiste di 4 problemi da risolvere ed assegna un voto massimo di 22 punti.

Il voto finale dell'esame è espresso in trentesimi ed è dato dalla somma dei voti delle due parti. Una somma superiore a 30 comporta l'assegnazione della lode.

9. programma esteso

Funzionamento del calcolatore: Architettura del calcolatore, sistema operativo, linguaggi di programmazione, compilazione ed esecuzione dei programmi, rappresentazione binaria dell'informazione.

Progettazione di algoritmi: modellazione di problemi, algoritmi, correttezza ed efficienza, algebra Booleana, diagrammi di flusso, istruzioni condizionali, istruzioni ripetitive, progettazione *top-down*, ricorsione.

Fondamenti di programmazione in Python: Variabili, espressioni e tipi numerici, aritmetica, istruzioni, *input-output*, istruzione *if-else*, ciclo *while*, ciclo *for*, funzioni.

Strutture dati: Stringhe, file, tuple, liste, dizionari.

Analisi e visualizzazione di dati: Librerie NumPy e Pandas.