

Testi del Syllabus

Resp. Did.	FANFANI MARCO	Matricola: 169319
Docente	FANFANI MARCO, 6 CFU	
Anno offerta:	2025/2026	
Insegnamento:	B034543 - FUNDAMENTALS OF COMPUTER PROGRAMMING	
Corso di studio:	B402 - ECONOMIA E COMMERCIO	
Anno regolamento:	2025	
CFU:	6	
Settore:	ING-INF/05	
Tipo Attività:	B - Caratterizzante	
Anno corso:	2	
Periodo:	II Semestre	



Testi in italiano

Lingua insegnamento	INGLESE
Obiettivi formativi	1. CONOSCENZA E COMPrensIONE 1.1 Conoscenza e comprensione delle nozioni di base di programmazione 1.2 Conoscenza e comprensione delle nozioni di base dei principali costrutti del linguaggio Python. 2. CAPACITA' DI APPLICARE CONOSCENZA E COMPrensIONE 2.1 Capacità di configurare un ambiente di sviluppo Python. 2.2. Capacità di scrivere script Python. 2.3 Capacità di programmazione orientata agli oggetti. 3. AUTONOMIA DI GIUDIZIO 3.1 Capacità di eseguire autonome scelte su script Python. 4. ABILITA' COMUNICATIVE 4.1 Capacità di gestione delle strutture di controllo e gestione degli errori 5. CAPACITA' DI APPRENDIMENTO 5.1 Capacità di reperire autonomamente documentazione tecnica su programmazione Python
Contenuti (Dipl.Sup.)	Concetti di base. IDE, interpreti e ambienti di sviluppo. Uso di pacchetti python. Uso di liste, tuple, dizionari, set. Controllo del Flusso. Utilizzo e definizione di funzioni. Importare e serializzare dati. Grafici e visualizzazioni di dati.
Prerequisiti	Nessuno
Metodi didattici	Il corso si compone di 48 ore di lezioni teoriche e pratiche in aula. Si consiglia l'uso di un proprio PC.
Modalità di verifica dell'apprendimento	L'esame consiste di una prova scritta e di una prova orale successiva alla prova scritta.
Programma esteso	Concetti di Base Introduzione alla programmazione. Caratteristiche di Python, Esecuzione di uno script Python, Sintassi di base: variabili, operatori, e tipi di dati IDE, Interpreti e Ambienti di Sviluppo Scelta e configurazione di un IDE (VSCode, Virtualenv, Jupyter Notebook), Utilizzo di interpreti Python, Creazione e gestione di ambienti virtuali, debugging Uso di Pacchetti Python Installazione di pacchetti con pip, Importazione di

moduli e pacchetti, Panoramica dei pacchetti più comuni: numpy, pandas, matplotlib Uso di Liste, Tuple, Dizionari, Set Condizioni (if, elif, else), Cicli (for, while), Interruzioni di ciclo (break, continue, pass) Utilizzo di Funzioni e Generatori Definizione di Funzioni: Sintassi delle funzioni, Argomenti posizionali e keyword, Argomenti di default, Argomenti arbitrari (*args, **kwargs) Funzioni Ricorsive: Concetto di ricorsione, Implementazione di funzioni ricorsive Generatori: Differenza tra generatori e funzioni normali, Utilizzo di yield, Generator expressions Importare e Serializzare Dati Lettura e Scrittura di File e introduzione ai principali formati: Gestione di file di testo, CSV, JSON Serializzazione: Introduzione a pickle, Salvataggio e caricamento di oggetti con pickle Grafici e Visualizzazioni di Dati Introduzione a matplotlib, Creazione di grafici base , Personalizzazione dei grafici (titoli, etichette, legende), Grafici statistici Programmazione ad Oggetti Concetti di Base: Classi e oggetti, Attributi e metodi Pillars della Programmazione ad Oggetti: Incapsulamento, Ereditarietà, Polimorfismo, Astrazione Metodi Speciali: Costruttori (__init__), Rappresentazioni stringa (__str__, __repr__)

Testi di riferimento

Slide, materiali e riferimenti web forniti a lezione.

Obiettivi Agenda 2030 per lo sviluppo sostenibile

Istruzione di Qualità. Lavoro dignitoso e crescita economica. Industria, innovazione e infrastrutture.

Obiettivi per lo sviluppo sostenibile

Codice	Descrizione
4	Istruzione di qualità
8	Lavoro dignitoso e crescita economica
9	Industria, innovazione e infrastrutture



Testi in inglese

Language	English
	1. KNOWLEDGE AND UNDERSTANDING 1.1 Knowledge and understanding of basic programming concepts. 1.2 Knowledge and understanding of fundamental constructs of the Python language. 2. ABILITY TO APPLY KNOWLEDGE AND UNDERSTANDING 2.1 Ability to configure a Python development environment. 2.2 Ability to write Python scripts. 2.3 Ability in object-oriented programming. 3.JUDGEMENT AUTONOMY 3.1 Ability to make autonomous choices regarding Python scripts. 4.COMMUNICATIVE SKILLS 4.1 Ability to manage control structures and handle errors. 5. LEARNING ABILITY 5.1 Ability to independently find technical documentation on Python programming
	Basic concepts. IDEs, interpreters, and development environments. Using Python packages. Using lists, tuples, dictionaries, and sets. Flow control. Usage and definition of functions. Importing and serializing data. Graphs and data visualizations.
	None
	The course consists of 48 hours of theoretical and practical in the classroom. Use of a personal PC is advised.

	The exam consists of a written test and an oral test following the written test.
	Basic Concepts Introduction to programming, Characteristics of Python, Running a Python Script, Basic Syntax: variables, operators, and data types IDE, Interpreters, and Development Environments Choosing and Configuring an IDE (VSCode, Virtualenv, Jupyter Notebook), Using Python Interpreters, Creating and Managing Virtual Environments, debugging Using Python Packages Installing Packages with pip, Importing Modules and Packages, Overview of Common Packages: numpy, pandas, matplotlib Using Lists, Tuples, Dictionaries, Sets Conditions: if, elif, else, Loops: for, while, Loop Interruptions: break, continue, pass Using Functions and Generators Defining Functions: function syntax, positional and keyword arguments, default arguments, arbitrary arguments (*args, **kwargs) Recursive Functions: concept of recursion, implementing recursive functions Generators: difference between generators and normal functions, using yield, generator expressions Importing and Serializing Data Reading and Writing Files and introduction to main formats: handling text CSV, and JSON files, Serialization: introduction to pickle, saving and loading objects with pickle, Graphs and Data Visualization Introduction to matplotlib, Creating Basic Charts, Customizing Charts: titles, labels, legends, Statistical Charts Object-Oriented Programming Basic Concepts: classes and objects, attributes and methods Pillars of Object-Oriented Programming: encapsulation, inheritance, polymorphism, abstraction Special Methods: constructors (init), string representations (str, repr)
	Slides, materials, and web references provided during the lecture.
	Quality Education. Decent work and economic growth. Industry, innovation and infrastructure.

Obiettivi per lo sviluppo sostenibile

Codice	Descrizione
4	Quality education
8	Decent work and economy growth
9	Industries, innovation and infrastructure