

CORSO DI STUDIO *Physics (LM-17)*
ANNO ACCADEMICO 2024-2025
DENOMINAZIONE DELL'INSEGNAMENTO *Advanced Programming in C++*

Principali informazioni sull'insegnamento	
Anno di corso	1°
Periodo di erogazione	2° semestre: Marzo - Maggio 2025
Crediti formativi universitari (CFU/ECTS):	3
SSD	FIS/01
Lingua di erogazione	Italiano ed Inglese
Modalità di frequenza	Obbligatoria

Docente	
Nome e cognome	Francesco S. Cafagna
Indirizzo mail	Francesco.cafagna@ba.infn.it
Telefono	+390805442328
Sede	INFN, sezione di Bari, via Orabona 4, 70125 Bari
Sede virtuale	
Ricevimento	Dalle 10:30 alle 11:30, Dipartimento di fisica, stanza 132 o per appuntamento.

Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (laboratorio, campo, esercitazione, altro)	Studio individuale
31	16	15	44
CFU/ECTS			
3	2	1	

Obiettivi formativi	Obiettivo principale del corso è introdurre gli studenti ai concetti fondanti dell'evoluzione dello stile di programmazione, da procedurale a quello orientato all'uso degli oggetti. Il linguaggio di programmazione C++, grazie alla sua natura generalista, al suo controllo della memoria ed all'intrinseco controllo della tipizzazione, verrà usato come paradigma di tale evoluzione; per questo motivo verranno trattate non solo le caratteristiche della grammatica di base, ma anche tutte quelle funzionalità che lo rendono il linguaggio di elezione per la semantica della programmazione orientata all'uso degli oggetti. Inoltre verrà offerta anche una panoramica delle novità introdotte delle più recenti standardizzazioni del linguaggio. Il piano delle lezioni prevede un certo numero di esercizi svolti, pensati per l'approfondimento degli argomenti teorici trattati.
Prerequisiti	Familiarità con personal computer. Conoscenza di base dei sistemi operativi più comuni: Linux/Unix, Windows, MAC OS.

Metodi didattici	Lezioni frontali con esercitazioni durante la lezione effettuate dal docente e dagli studenti, utilizzando sia ambienti di programmazione virtuali, ospitati su un sito di <i>e-learning</i> dedicato, sia su <i>personal computer</i> .
-------------------------	--

Risultati di apprendimento previsti	Conoscenza della programmazione procedurale ed orientata agli oggetti. Capacità di creare software personalizzato e dedicato alla soluzione dei problemi connessi ad analisi di dati o gestione di apparati di misura.
--	---

<p>Da indicare per ciascun Descrittore di Dublino (DD=</p> <p>DD1 Conoscenza e capacità di comprensione</p> <p>DD2 Conoscenza e capacità di comprensione applicate</p> <p>DD3-5 Competenze trasversali</p>	<p>- Descrittore di Dublino 1: <i>conoscenza e capacità di comprensione (che cosa lo/la studente/studentessa conosce al termine dell'insegnamento);</i></p> <ul style="list-style-type: none"> o Conoscenza del linguaggio di programmazione C++. o Conoscenza di un linguaggio di programmazione orientato ad oggetti. o Capacità di comprensione di un codice scritto in un linguaggio di programmazione procedurale ed orientato ad oggetti. o Conoscenza degli strumenti necessari per creare un programma originale su un personal computer. <p>- Descrittore di Dublino 2: <i>capacità di applicare conoscenza e comprensione (che cosa lo/la studente/studentessa sa fare al completamento dell'insegnamento ovvero quali sono le competenze che ha acquisito);</i></p> <ul style="list-style-type: none"> o Capacità di sviluppo di qualsiasi tipo di algoritmo su qualsiasi piattaforma di calcolo per qualsiasi tipo di applicazione. o Capacità di creare software personalizzato e dedicato alla soluzione dei problemi connessi ad analisi di dati o gestione di apparati di misura. <p>- Descrittore di Dublino 3: <i>capacità critiche e di giudizio (occorre indicare le attività che concorrono allo sviluppo di tali abilità. Per es.: prove di laboratorio, redazione di relazioni scritte, e così via); Gli/Le studenti/studentesse devono avere la capacità di raccogliere ed interpretare i dati (normalmente nel proprio campo di studio) ritenuti utili a determinare giudizi autonomi, inclusa la riflessione su temi sociali, scientifici o etici ad essi connessi.</i></p> <ul style="list-style-type: none"> • Autonomia di giudizio <p>Al termine dell'insegnamento lo/la studente/studentessa sarà in grado di:</p> <ul style="list-style-type: none"> o valutare le migliori soluzioni di disegno, sintassi del linguaggio di programmazione, e strumenti di libreria da utilizzare per realizzare nel modo più efficiente il compito da fare eseguire ad un programma. <p>- Descrittore di Dublino 4: <i>capacità di comunicare quanto si è appreso (anche in questo caso si devono predisporre attività mirate allo sviluppo, nello/a studente/studentessa, della capacità di comunicare/trasmettere quanto appreso); gli studenti devono saper comunicare informazioni, idee, problemi e soluzioni a interlocutori specialisti e non specialisti.</i></p> <ul style="list-style-type: none"> • Abilità comunicative <p>Al termine dell'insegnamento lo/la studente/studentessa sarà in grado di:</p> <ul style="list-style-type: none"> o Apprendere la semantica di un linguaggio di programmazione e descrivere la struttura di un codice con termini appropriati. <p>- Descrittore di Dublino 5: <i>capacità di proseguire lo studio in modo autonomo nel corso della vita (occorre indicare quali siano gli strumenti forniti affinché lo studente sappia, al termine dell'insegnamento, proseguire autonomamente nello studio). Gli/Le studenti/studentesse devono aver sviluppato quelle capacità di apprendimento che sono loro necessarie per intraprendere studi successivi con un alto grado di autonomia.</i></p> <ul style="list-style-type: none"> • Capacità di apprendere in modo autonomo <p>Al termine dell'insegnamento lo/la studente/studentessa sarà in grado di:</p> <ul style="list-style-type: none"> • Utilizzare le fonti e gli strumenti di riferimento per aggiornarsi sull'evoluzione del linguaggio di programmazione. • Creare programmi originali da utilizzare per la soluzione di problemi legati alle proprie attività di ricerca.
<p>Contenuti di insegnamento (Programma)</p>	<p>Introduzione:</p> <ul style="list-style-type: none"> - Dallo stile di programmazione procedurale a quello orientato all'uso degli oggetti. - Programmazione: una introduzione generale. - Programmazione: concetti base. - Programmazione: il gergo.

	<ul style="list-style-type: none"> - Introduzione all'uso essenziale degli strumenti necessari alla creazione di un eseguibile. - Un linguaggio di programmazione orientate all'uso degli oggetti: C++. <p>La grammatica di base del C++:</p> <ul style="list-style-type: none"> - I tipi base. - Expressions e statements. - Funzioni. - Puntatori e reference. - Esempi ed esercizi. <p>Funzionalità avanzate del C++:</p> <ul style="list-style-type: none"> - Tipi aggregati: Structure e Union. - Function overloading. - Namespaces. - Template programming. - Esempi ed esercizi. <p>La classe:</p> <ul style="list-style-type: none"> - Introduzione e proprietà generali. - Class members: creatori, distruttori, metodi ed helper functions. - Controllo dell'accesso ai membri della classe: public, private. - Operatori ed overloading. - Esempi ed esercizi. <p>Le classi, funzionalità avanzate:</p> <ul style="list-style-type: none"> - Classi derivate. - Ereditarietà e polimorfismo. - Esempi ed esercizi. <p>La Standard Template Library: STL.</p> <ul style="list-style-type: none"> - Una introduzione e proprietà generali. - Approfondimento dei tipi maggiormente utilizzati: string, containers, functionals ed algorithms.
Testi di riferimento	<p>B. Stroustrup, The C++ programming language (Third edition), Addison – Wesley.</p> <p>B. Stroustrup, Programming -- Principles and Practice Using C++, Addison -Wesley ISBN 978-0321543721. December 2008.</p> <p>S. Lippman et al., C++ Primer (Fifth edition), Addison – Wesley.</p> <p>S. Meyer, Effective C++ Third Edition, Addison-Wesley ISBN-13: 978-0321334879 (any edition of this book is worth consulting).</p> <p>S. Meyer, Effective Modern C++, O'Reilly Media, Incorporated; ISBN-13: 978-1491903995.</p> <p>H. Sutter, Exceptional C++: 47 engineering Puzzles, Programming Problems and Solutions. Reading, MA: Addison-Wesley.</p>
Note ai testi di riferimento	
Materiali didattici	Il materiale didattico è disponibile online su una piattaforma e-learning (MOODLE).

Valutazione	
Modalità di verifica dell'apprendimento	Scritto (realizzazione di un programma originale o modifica di un modello schematico) e orale.
Criteri di valutazione	<ul style="list-style-type: none"> ● <i>Conoscenza e capacità di comprensione:</i> <ul style="list-style-type: none"> o valutazione della conoscenza della sintassi di base del linguaggio di programmazione. ● <i>Conoscenza e capacità di comprensione applicate:</i> <ul style="list-style-type: none"> o valutazione della capacità di applicazione delle tecniche di programmazione apprese, ad un problema semplice. ● <i>Autonomia di giudizio:</i>

	<ul style="list-style-type: none"> o capacità di discernere tra le differenti tecniche e caratteristiche del linguaggio e valutarne l'applicabilità ai vari casi di studio ● Abilità comunicative: <ul style="list-style-type: none"> o saper descrivere in maniera chiara e sintetica il proprio codice, le scelte di disegno fatte ed il comportamento dei vari componenti del programma. ● Capacità di apprendere: <ul style="list-style-type: none"> o valutazione della modalità di utilizzo del linguaggio di programmazione e degli argomenti affrontati a lezione nel creare frammenti di codice o programmi.
<p>Criteria di misurazione dell'apprendimento e di attribuzione del voto finale</p>	<p>Il voto finale è attribuito in trentesimi. L'esame si intende superato quando il voto è maggiore o uguale a 18.</p> <p>La prova scritta, a seconda della complessità dell'esercizio proposta, sarà valutata con un massimo di 20 trentesimi.</p> <p>La prova orale sarà valutata con un massimo di 10 trentesimi.</p> <p>Durante la prova orale verrà particolarmente valutata la capacità di giudizio nella scelta delle soluzioni di disegno del codice in funzione dell'efficienza di esecuzione del codice, nonché la capacità di esposizione in termini appropriati e chiari.</p>
<p>Altro</p>	
	.