

FACOLTÀ	Ingegneria
ANNO ACCADEMICO	2014/2015
CORSO DI LAUREA	Ingegneria Informatica e delle Telecomunicazioni
INSEGNAMENTO	Programmazione
TIPO DI ATTIVITÀ	Caratterizzante
AMBITO DISCIPLINARE	Ingegneria Informatica
CODICE INSEGNAMENTO	05871
ARTICOLAZIONE IN MODULI	NO
NUMERO MODULI	
SETTORI SCIENTIFICO DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	Marco La Cascia Professore Associato Università di Palermo
CFU	9
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	141
NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE	84
PROPEDEUTICITÀ	Calcolatori elettronici
ANNO DI CORSO	II
SEDE DI SVOLGIMENTO DELLE LEZIONI	Consultare il sito www.ingegneria.unipa.it
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali Esercitazioni teoriche Esercitazioni in aula informatica
MODALITÀ DI FREQUENZA	Facoltativa
METODI DI VALUTAZIONE	Discussione sugli argomenti del corso Sviluppo di un programma completo in C++
TIPO DI VALUTAZIONE	Voto in trentesimi
PERIODO DELLE LEZIONI	Primo semestre
CALENDARIO DELLE ATTIVITÀ DIDATTICHE	Consultare il sito www.ingegneria.unipa.it
ORARIO DI RICEVIMENTO DEGLI STUDENTI	Martedì 15:00-17:00

<p>RISULTATI DI APPRENDIMENTO ATTESI</p> <p>Conoscenza e capacità di comprensione Lo studente al termine del Corso avrà conoscenza delle problematiche inerenti la programmazione e la progettazione ad oggetti e lo sviluppo e manutenzione di software scritto in C++. In particolare lo studente conoscerà nel dettaglio caratteristiche del C++ come operatori, puntatori, funzioni, classi e oggetti, ereditarietà, funzioni virtuali, polimorfismo, sistema di I/O, namespace, gestione delle eccezioni. Per il raggiungimento di questo obiettivo il corso comprende: lezioni frontali; analisi e discussione di frammenti di programmi. Per la verifica di questo obiettivo l'esame comprende la discussione sugli argomenti del programma.</p> <p>Capacità di applicare conoscenza e comprensione Lo studente sarà in grado di progettare e sviluppare software complessi utilizzando le</p>
--

caratteristiche avanzate del C++; saprà intervenire su software esistente al fine di correggerne o incrementarne le funzionalità.

Per il raggiungimento di questo obiettivo il corso comprende: esercitazioni teoriche; esercitazioni da svolgere autonomamente.

Per la verifica di questo obiettivo l'esame comprende la scrittura di un programma completo in C++ a partire dalla descrizione testuale del problema da risolvere.

Autonomia di giudizio

Lo studente sarà in grado di analizzare classi e librerie di terze parti e valutarne la possibile adozione nello sviluppo di software complessi.

Per il raggiungimento di questo obiettivo il corso comprende: analisi e discussioni su casi di studio; una presentazione sommaria della libreria standard e delle principali strutture dati e discussioni su possibili vantaggi e svantaggi derivanti dal loro uso.

Per la verifica di questo obiettivo l'esame comprende la scrittura di un programma completo in C++ in cui lo studente è obbligato a effettuare delle scelte progettuali in autonomia.

Abilità comunicative

Lo studente acquisirà la capacità di comunicare ed esprimere problematiche inerenti l'oggetto del corso. Sarà in grado di sostenere conversazioni su tematiche di sviluppo software, di evidenziare problemi relativi a progetto e implementazione e di offrire soluzioni.

Per il raggiungimento di questo obiettivo il corso comprende: esercitazioni in aula informatica in cui gli studenti espongono come risolvono gli esercizi e le eventuali difficoltà incontrate.

Per la verifica di questo obiettivo l'esame comprende un esame orale sugli argomenti del corso e la discussione sul programma in C++ scritto.

Capacità d'apprendimento

Lo studente sarà in grado di affrontare in autonomia qualsiasi problematica relativa allo sviluppo di software con linguaggi ad oggetti. Sarà in grado di approfondire tematiche complesse quali il polimorfismo, la gestione dinamica della memoria, la scalabilità del software, etc...

Per il raggiungimento di questo obiettivo il corso comprende: esercitazioni da svolgere autonomamente; discussione sulle eventuali difficoltà incontrate.

Per la verifica di questo obiettivo l'esame comprende la discussione su alcuni argomenti introdotti a lezione e il cui approfondimento è lasciato agli studenti.

OBIETTIVI FORMATIVI

Il corso tratta in maniera approfondita la programmazione a oggetti in C++. Verranno trattati sia gli aspetti relativi alla programmazione ad oggetti in generale che aspetti peculiari del C++ quali l'overloading degli operatori, la gestione di memoria dinamica, i namespace, i template e la libreria standard. Verranno infine trattate le strutture dati di uso più frequente inquadrando nel rigore formale del C++.

	PROGRAMMAZIONE
ORE FRONTALI	LEZIONI FRONTALI
2	Introduzione all'informatica e alla programmazione: rappresentazione dell'informazione nei computer, concetto di algoritmo, programmazione strutturata e orientata agli oggetti, linguaggi di programmazione C e C++, linguaggio di modellazione UML 2.0.
4	Il linguaggio C++, elementi base: costruzione di un programma in C++, struttura generale di un programma C++, creazione, esecuzione e debugging di un programma in C++, elementi di un programma in C++, tipi di dato in C++, costanti e variabili, durata e visibilità di una variabile, istruzione di assegnamento, I/O da console.
3	Operatori ed espressioni: concetti generali, operatore di assegnamento, operatori aritmetici, di incremento e decremento, relazionali, logici, di manipolazione dei bit, operatore condizionale, operatore virgola, operatore sizeof, conversioni di tipo.
3	La programmazione strutturata: strutture di controllo, le istruzioni if e if else, istruzioni if else annidate, l'istruzione switch, l'istruzione while, il ciclo for, il ciclo do while, cicli annidati.

4	Funzioni: concetto di funzione, struttura di una funzione, prototipi delle funzioni, passaggio di parametri a una funzione, argomenti di default, funzioni inline, regole di visibilità e classi di immagazzinamento, visibilità di una funzione, concetto e uso di funzioni di libreria, funzioni di libreria comuni, compilazione modulare, sovraccaricamento di funzioni, ricorsione, template di funzioni.
2	Array: concetto di array, inizializzazione di array, array di caratteri, array multidimensionali, passaggi di vettori come parametri di funzione.
2	Strutture e unioni: concetto di struttura, accesso ai campi delle strutture, strutture annidate, array di strutture, utilizzare le strutture come parametri, funzioni membri di strutture, unioni, enumerazioni, typedef.
3	Puntatori e riferimenti: concetto di riferimento, concetto di puntatore, puntatori null, puntatore a puntatore, puntatori ed array, array di puntatori, puntatori e stringhe, aritmetica dei puntatori, puntatori costanti e puntatori a costanti, puntatori come argomenti di funzioni, puntatori a strutture.
1	Allocazione dinamica della memoria: gestione dinamica della memoria, operatori new e delete, tipi di memoria in C++.
2	Stringhe: concetto di stringa, lettura di stringhe, array e stringhe come parametri di funzione, la libreria cstring, la classe string della libreria standard, string stream.
5	Classi e oggetti: concetto di classe e oggetto, definizione di una classe, costruttori, distruttori, overloading di funzioni membro.
5	Classi derivate, ereditarietà e polimorfismo: classi derivate, tipi di ereditarietà, costruttori, distruttori, binding, funzioni virtuali, polimorfismo, vantaggi del polimorfismo.
3	Flussi file e la libreria standard di I/O: concetto di stream, la libreria di classi per l'I/O, le classi istream e ostream, formattazione dell'output, indicatori di formato, I/O da file, I/O binario, accesso diretto.
3	Overloading degli operatori: concetto di overloading, overloading di operatori unari e binari, overloading degli operatori + e -, overloading dell'operatore di assegnamento, overloading degli operatori << e >>.
1	Eccezioni: condizioni di errore nei programmi, gestione delle eccezioni in C++, specifica delle eccezioni.
2	Liste, pile e code: concetto di lista, operazioni con le liste, lista doppiamente concatenata, liste circolari, concetto e gestione di una pila, concetto e gestione di una coda. Alberi: concetto di albero, alberi binari, struttura di un albero binario, visita di un albero.
1	Template: template in C++, template di funzioni e di classi, compilazione di template.
2	Strutture dati nella libreria standard del C++.
ESERCITAZIONI	
6	Ambiente di sviluppo e compilazione. Semplici programmi in C++.
3	Programmazione strutturata in C++
3	Funzioni.
3	Programmi che utilizzano array e strutture.
6	Uso dei puntatori e della memoria dinamica.
6	Implementazione di classi.
3	Gerarchie di classi e polimorfismo.
3	Overloading di operatori.
3	Sviluppo di prove di esami di anni precedenti.
TESTI CONSIGLIATI	S.B. LIPPMAN, J. LAJOIE, B.E. MOO (2012). <i>C++ Primer 5th ed.</i> Addison Wesley.
	LUIS JOYANES AGUILAR (2008). <i>Fondamenti di programmazione in C++</i> . McGraw-Hill.
	H.M. DEITEL, P.J. DEITEL (2005). <i>C++ Fondamenti di programmazione 2a ed.</i> Apogeo.