



UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria
ACADEMIC YEAR	2015/2016
BACHELOR'S DEGREE (BSC)	COMPUTER AND TELECOMMUNICATION ENGINEERING
SUBJECT	PROGRAMMING
TYPE OF EDUCATIONAL ACTIVITY	B
AMBIT	50289-Ingegneria informatica
CODE	05871
SCIENTIFIC SECTOR(S)	ING-INF/05
HEAD PROFESSOR(S)	LO PRESTI LILIANA Professore Associato Univ. di PALERMO LA CASCIA MARCO Professore Ordinario Univ. di PALERMO
OTHER PROFESSOR(S)	
CREDITS	9
INDIVIDUAL STUDY (Hrs)	144
COURSE ACTIVITY (Hrs)	81
PROPAEDEUTICAL SUBJECTS	
MUTUALIZATION	
YEAR	2
TERM (SEMESTER)	1° semester
ATTENDANCE	Not mandatory
EVALUATION	Out of 30
TEACHER OFFICE HOURS	LA CASCIA MARCO Monday 15:00 17:00 Microsoft Teams Codice: wztkv0u LO PRESTI LILIANA Tuesday 16:00 17:00

PREREQUISITES	
LEARNING OUTCOMES	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 Java. In particolare lo studente conoscerà nel dettaglio caratteristiche del linguaggio Java come operatori, funzioni, strutture di controllo, classi e oggetti, ereditarietà, polimorfismo, interfacce, metodi e classi generiche, gestione delle eccezioni, collezioni, sistema di I/O. 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. Capacità di applicare conoscenza e comprensione Lo studente sarà in grado di progettare e sviluppare software complessi utilizzando le 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. Sarò 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.
ASSESSMENT METHODS	Sviluppo di un programma completo in Java. Discussione sugli argomenti del corso.
EDUCATIONAL OBJECTIVES	Il corso tratta in maniera approfondita la programmazione a oggetti in Java. Verranno trattati sia gli aspetti relativi alla programmazione ad oggetti in generale che aspetti peculiari del linguaggio Java quali l'organizzazione del codice in package, la documentazione del codice, le funzionalità per la programmazione concorrente e la libreria standard. Verranno infine trattate le strutture dati di uso più frequente inquadrando nel contesto delle librerie Java.
TEACHING METHODS	Lezioni frontali, esercitazioni in aula informatica
SUGGESTED BIBLIOGRAPHY	CLAUDIO DE SIO CESARI (2014). Manuale di Java 8. Programmazione orientata agli oggetti con Java standard edition 8. Hoepli PELLEGRINO PRINCIPE (2014). Java 8. Apogeo. PATRICK NIEMEYER. Learning Java 4th edition. O'Really

SYLLABUS

Hrs	Frontal teaching
2	Introduzione a Java

SYLLABUS

Hrs	Frontal teaching
2	Componenti fondamentali di un programma Java: le basi della programmazione object-oriented, i metodi in Java, le variabili, i metodi costruttori, i package.
4	Identificatori, tipi di dati e array: i tipi di dati primitivi e non primitivi, gli array in Java.
4	Operatori e gestione del flusso di esecuzione: gli operatori di base, la gestione del flusso di esecuzione, costrutti di programmazione semplici e avanzati.
6	Incapsulamento e visibilità: i paradigmi della programmazione a oggetti, l'incapsulamento, i modificatori d'accesso, il modificatore static.
4	Ereditarietà e interfacce: l'ereditarietà, il modificatore final, la classe Object, ereditarietà e incapsulamento, le interfacce.
4	Polimorfismo: il polimorfismo per metodi, overload e override, il polimorfismo per dati.
4	Eccezioni e asserzioni: il meccanismo per la gestione delle eccezioni in Java, la propagazione dell'eccezione, introduzione alle asserzioni.
2	Enumerazioni e tipi innestati: le classi interne, le classi anonime, i tipi enumerazione.
2	Tipi generici: creare propri tipi generici, la deduzione automatica del tipo.
4	Libreria: le classi String, Object, System, Runtime e Math, il package java.util, Date-Time API.
2	Gestione dei thread: la classe Thread, la sincronizzazione di thread, la comunicazione fra thread, la gestione della concorrenza.
4	Collezioni: interfacce Collection, List, Queue e Deque, Map e SortedMap, algoritmi di utilità.
2	Input/Output: character e byte stream, la lettura di input da tastiera, la gestione dei files, la serializzazione di oggetti.
2	Interfacce grafiche: JavaFX, la creazione di interfacce con i Layout, la gestione degli eventi.
Hrs	Practice
6	Ambiente di sviluppo e compilazione. Semplici programmi in Java.
3	Implementazione di semplici algoritmi in Java.
3	Implementazione di programmi che utilizzano array.
6	Implementazione di semplici classi.
6	Implementazione di gerarchie di classi e polimorfismo.
3	Utilizzo di enumerazioni e tipi generici.
3	Uso della libreria di Java
3	Programmi che utilizzano collezioni e il sistema di I/O.
3	Sviluppo di prove di esami di anni precedenti.