

Programmazione II (Resto 1)

a.a. 2016/2017

Introduzione al corso

Prof. Carmine Gravino
Email: gravino@unisa.it

Informazioni generali

- **Crediti:** 10 CFU
 - 56 ore Frontali
 - 30 ore di Laboratorio/Esercitazioni
- **Orario lezioni:**
 - Lunedì 15-17 (aula F8)
 - Martedì 13-16 (Lab. Turing)
 - Venerdì 14-17 (aula P4)
- **Sito del corso:**
<http://elearning.informatica.unisa.it/el-platform/>
- **Testo:** Concetti di Informatica e fondamenti di Java (versioni 5, 6, e 7) Cay Horstmann. Apogeo

Informazioni generali



- **Orario ricevimento:**
 - Lunedì 17-18
 - Mercoledì 15-17
 - Venerdì 17-18
 - (contattare il docente via email)

CORSI II ANNO - CLASSE 2 - MATRICOLA CONGRUO A 1

I Semestre - Anno Accademico 2016-2017

ORARIO	LUNEDI'	MARTEDI'	MERCOLEDI'	GIOVEDI'	VENERDI'
09 - 10	SO - Lab Sistemi	BD - Lab Sistemi		SO - Lab Sistemi	SO - Lab Sistemi
10 - 11	SO - Lab Sistemi	BD - Lab Sistemi		SO - Lab Sistemi	SO - Lab Sistemi
11 - 12	SO - Lab Sistemi	BD - Lab Sistemi		SO - Lab Sistemi	BD - P/4
12 - 13				CPSM - P/4	BD - P/4
13 - 14	CPSM - F/8	Prog II - Lab Turing		CPSM - P/4	
14 - 15	CPSM - F/8	Prog II - Lab Turing			Prog II - P/4
15 - 16	Prog II - F/8	Prog II - Lab Turing	Ricevimento	BD - P/4	Prog II - P/4
16 - 17	Prog II - F/8		Ricevimento	BD - P/4	Prog II - P/4
17 - 18	Ricevimento				Ricevimento

Leggenda
SO = Sistemi Operativi (CARPENTIERI)
Prog II = Programmazione II (GRAVINO)
CPSM = Calcolo Probabilità e Statistica Matematica (DI CRESCENZO - MARTINUCCI)
BD = Basi di Dati (POLESE)
Ore di recupero Lab Turing

Informazioni generali (cont.)

- **Durata del corso:**
 - Martedì 22 Settembre – Mercoledì 21 Dicembre 2016
- **Prima prova:**
 - Seconda settimana Novembre 2016
- **Seconda prova:**
 - Ultima settimana del corso
- **Preappello:** 10 gennaio
- **I appello:** 27 gennaio
- **II appello:** 15 febbraio
(salvo modifiche degli uffici)

Informazioni generali (cont.)

- **Verifica dell'apprendimento:**
 - gli studenti devono superare un esame finale costituito da una prova pratica in laboratorio ed una prova orale.
 - Una tipica prova pratica richiede la soluzione di esercizi di programmazione in java e la loro implementazione con l'ausilio di un pc, utilizzando l'ambiente integrato di sviluppo software appreso durante il corso.
 - La prova orale consiste in un colloquio sui progetti discussi al corso, gli esercizi svolti nella prova pratica e gli argomenti in programma.

Syllabus

- **Prerequisiti:** Lo studente dovrebbe avere una buona conoscenza del linguaggio C, delle tecniche di programmazione procedurale e delle problematiche relative alla codifica in codice di algoritmi semplici. Conoscenze di base e discreta familiarità con un sistema operativo (windows, linux,...).

Syllabus (cont.)

- **Conoscenza e capacità di comprensione**
 - acquisizione delle competenze di base per la progettazione object-oriented attraverso la comprensione dei concetti di astrazione sui dati, di incapsulamento dell'informazione, di coesione e accoppiamento, e di riutilizzo del codice. Comprendere le differenze tra questo paradigma e il paradigma procedurale. Sintassi completa del linguaggio di programmazione java. Utilizzo del linguaggio java per la definizione di classi e per la promozione del riutilizzo del software. Conoscenza delle principali classi messe a disposizione dall'ambiente jdk 7. Conoscenza dei fondamenti teorici del polimorfismo e del meccanismo dell'ereditarietà. Conoscenza e padronanza dei concetti di classi astratte, interfacce e del meccanismo di gestione delle eccezioni. Rudimenti di programmazione grafica e ad eventi in java. Realizzazione e consultazione di documentazione di un programma.

Syllabus (cont.)

- **Capacità di applicare conoscenza e comprensione:**
 - capacità di analisi di problemi, di specifica dei requisiti e di definizione di una strategia risolutiva con un approccio orientato agli oggetti, con la sua implementazione nel linguaggio java, garantendo il giusto equilibrio tra qualità ed efficienza del software. Sviluppo di progetti con attività individuali e di gruppo.
- **Autonomia di giudizio:**
 - discussione dei pro e contro delle tecniche e soluzioni studiate. Analisi critica dei lavori progettuali.

Syllabus (cont.)

- **Abilità comunicative:**
 - partecipazione alla formulazione delle strategie risolutive rispetto ai problemi proposti. Presentazione e discussione delle soluzioni individuate.
- **capacità di apprendere:**
 - consultazione individuale della documentazione delle librerie java. Ricerca di classi esistenti e individuazione di nuovi metodi per la soluzione dei problemi proposti.

Contenuti del corso (cont.)

- Introduzione alla programmazione orientata agli oggetti e ai concetti di astrazione dei dati e di incapsulamento.
- Introduzione a java e al processo di compilazione.
- Introduzione al toolkit di programmazione jdk 7.0.
- Oggetti, variabili, riferimenti; classi e metodi, comunicazione fra oggetti, parametri espliciti e impliciti; il riferimento this.
- Tipi di dati fondamentali.
- Aritmetica e funzioni matematiche.
- Invocazione di metodi statici.
- La classe string; l'input da finestra di dialogo.
- Imparare ad acquisire dati in ingresso da console; conoscere il tipo di dati char; l'input da file.
- File e flussi. Leggere e scrivere file di testo; formato binario e formato di testo

Contenuti del corso (cont.)

- Utilizzo degli array in java. Richiami sui costrutti di controllo.
- Le collezioni in java: l'uso di vettori (array lists) come raccoglitori di oggetti; comprendere quando è il caso di usare vettori e quando array nei nostri programmi; le classi involucro.
- Cenni sui tipi generici, sui tipi enumerativi e sui meccanismi di 'autoboxing' e 'unboxing'.
- Progettazione di classi. Imparare a scegliere le classi appropriate da implementare comprendere i concetti di coesione e accoppiamento; minimizzare l'uso degli effetti collaterali comprendere la differenza tra metodi di istanza e metodi statici; comprendere le regole di portata delle variabili locali e delle variabili istanza; chiamata per valore e chiamata per riferimento; documentare le responsabilità dei metodi e dei loro chiamanti con precondizioni e postcondizioni.

Contenuti del corso (cont.)

- Cenni sull'uso delle schede CRC nella progettazione delle classi. Pacchetti.
- Imparare cosa sono le interfacce; essere in grado di convertire tra riferimenti a classe e a interfacce; comprendere il concetto di polimorfismo; apprezzare come le interfacce possano essere usate per ridurre l'accoppiamento tra le classi; le classi interne.
- Ereditarietà e riuso. Comprendere la relazione di ereditarietà tra classi; la riscrittura di metodi; la ridefinizione di campi; la conversione di tipo tra sottoclasse e superclasse.
- Il polimorfismo e il binding dinamico; le classi astratte; il controllo di accesso; la superclasse universale object.
- Meccanismo di gestione delle eccezioni. Imparare cosa sono le eccezioni controllate e non controllate; il ciclo try-catch-finally; progettare le eccezioni.

Contenuti del corso (cont.)

- Testing: verifica del comportamento degli oggetti; scelta dei casi di test; Integrazione di classi.
- Programmazione grafica e applet.
- Introduzione alle applet e alla programmazione grafica: gestione delle forme, dei font e dei colori; utilizzo di componenti per l'input; primitive per la gestione del contesto grafico; personalizzazione dei pannelli, gestione del layout, gestione dei principali componenti swing.
- Programmare ad eventi: comprendere il modello a eventi di java; installare (o registrare) listener di mouse o di azioni.
- Cenni sul paradigma model-view-controller per lo sviluppo di interfacce grafiche

Workplan del Corso

	Lunedì (ore: 15-17 aula: P4)	Martedì (ore: 13-16 Lab: Turing)	Venerdì (ore: 14-17 aula: P4)
Settimana 1 (22-23 settembre)			Introduzione al corso
Settimana 2 (26-30 settembre)	Programmazione OO, Java		Programmazione OO, Java
Settimana 3 (3-7 ottobre)	Utilizzare gli Oggetti		Realizzare Classi
Settimana 4 (10-14 ottobre)	Tipi di Dati Fondamentali		Decisioni/Iterazioni
Settimana 5 (17-21 ottobre)	Vettori e Array		Progettazione di Classi
Settimana 6 (24-28 ottobre)	Progettazione OO		Interfacce e Polimorfismo
Settimana 7 (31 ott -4 novembre)	Interfacce e Polimorfismo		Ereditarietà
Settimana 8 (7-11 novembre)	Ereditarietà		1^ Intercorso
Settimana 9 (14-18 novembre)	Testing e Debugging		Gestione delle Eccezioni
Settimana 10 (21-25 novembre)	Gestione delle Eccezioni		IO e Flussi
Settimana 11 (28 nov. - 2 dicembre)	Programmazione Grafica		Gestione degli Eventi
Settimana 12 (5-9 dicembre)	Gestione degli Eventi		Interfacce Grafiche Utente
Settimana 13 (12-16 dicembre)	Programmazione Generica - Thread		2^ Intercorso
Settimana 14 (19-21 dicembre)			