

Testi del Syllabus

Resp. Did.	POGGI Agostino	Matricola: 004617
Anno offerta:	2016/2017	
Insegnamento:	1006642 - PARADIGMI E LINGUAGGI DI PROGRAMMAZIONE	
Corso di studio:	5015 - INGEGNERIA INFORMATICA	
Anno regolamento:	2016	
CFU:	6	
Settore:	ING-INF/05	
Tipo Attività:	D - A scelta dello studente	
Anno corso:	1	
Periodo:	II° semestre	
Sede:	PARMA	



Testi in italiano

Lingua insegnamento	Italiano
Contenuti	<ol style="list-style-type: none">1. Introduzione ai linguaggi formali2. Programmazione ad oggetti3. Programmazione generica4. Programmazione funzionale5. Linguaggi dinamici e metaprogrammazione
Testi di riferimento	<ul style="list-style-type: none">• M. Gabbrielli, S. Martini (2011). Linguaggi di programmazione - Principi e paradigmi. 9788838665738.• M. Lipovaca (2012). Learn You a Haskell for Great Good. 9781593272838. http://learnyouahaskell.com/• S. B. Lippman, J. Lajole, B. E. Moo (2013). C++ Primer. 9780321714111
Obiettivi formativi	<p>L'obiettivo del corso è fornire allo studente la capacità di comprendere i principi dei linguaggi di programmazione e le tecniche di programmazione moderna secondo i diversi principali paradigmi:</p> <ul style="list-style-type: none">• Basi di teoria dei linguaggi e dei compilatori• Programmazione ad oggetti• Programmazione generica• Programmazione funzionale <p>Le capacità di applicare le conoscenze elencate riguardano soprattutto lo sviluppo multi-paradigma, adatto al contesto di sistemi distribuiti, eterogenei e paralleli:</p> <ul style="list-style-type: none">• Gestire in maniera corretta dati e codice in forma testuale• Riconoscere ed utilizzare caratteristiche comuni a vari linguaggi• Utilizzare l'analisi statica per ottenere codice robusto e performante• Utilizzare il polimorfismo e la metaprogrammazione per alzare il livello di astrazione• Utilizzare la programmazione funzionale per l'integrità e la computazione parallela
Prerequisiti	Nessuna propedeuticità. Si suppone comunque che lo studente abbia buone basi di programmazione strutturata e ad oggetti.

Metodi didattici	Lezioni in aula, con l'ausilio di slide rese disponibili in anticipo agli studenti. Possibilità di adottare l'approccio Flipped Classroom e seminari per alcuni contenuti. Esercizi di programmazione in laboratorio.
Altre informazioni	<p>Testi alternativi e di approfondimento</p> <ul style="list-style-type: none"> • J. E. Hopcroft (2009). Automi, linguaggi e calcolabilità. 9788871925523. • A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman (2009). Compilatori - Principi, tecniche e strumenti. 9788871925592. • B. Stroustrup (2015). C++ - Linguaggio, libreria standard, principi di programmazione. 9788865184486.
Modalità di verifica dell'apprendimento	L'esame consiste di una prova scritta, una prova di programmazione, che può essere svolta anche durante le esercitazioni, e la presentazione di un progetto finale, che applichi lo sviluppo multi-paradigma ad un caso reale.
Programma esteso	<ol style="list-style-type: none"> 1. Introduzione ai linguaggi formali (12 ore in aula e 4 in lab) <ol style="list-style-type: none"> a. Classificazione di Chomsky b. Espressioni regolari c. Grammatiche non contestuali d. Analisi sintattica e generatori di parser e. Alberi sintattici e generazione del codice 2. Programmazione ad oggetti (4 ore in aula e 2 in lab) <ul style="list-style-type: none"> . Introduzione a. Programmazione ad oggetti in C++14 b. Confronto con altri linguaggi: Java, Python, Go 3. Programmazione generica (4 ore in aula e 2 in lab) <ul style="list-style-type: none"> . Introduzione a. Template C++ e STL 4. Paradigma di programmazione funzionale (8 ore in aula e 4 in lab) <ul style="list-style-type: none"> . Introduzione a. Presentazione del linguaggio LISP b. Programmazione in Haskell c. Caratteristiche funzionali del C++14 d. Programmazione parallela con il C++14 5. Linguaggi dinamici e metaprogrammazione (4 ore in aula e 2 in lab) <ul style="list-style-type: none"> . Introduzione a. Caratteristiche dinamiche del C++14 b. Linguaggi dinamici e di scripting c. Metaprogrammazione in Python



Testi in inglese

Lingua insegnamento	Italian
Contenuti	<ol style="list-style-type: none"> 1. Introduction to formal languages 2. Object-oriented programming 3. Generic programming 4. Functional programming 5. Dynamic languages and metaprogramming
Testi di riferimento	<ul style="list-style-type: none"> • M. Gabbrielli, S. Martini (2011). Linguaggi di programmazione - Principi e paradigmi. 9788838665738. • M. Lipovaca (2012). Learn You a Haskell for Great Good. 9781593272838. http://learnyouahaskell.com/ • S. B. Lippman, J. Lajole, B. E. Moo (2013). C++ Primer. 9780321714111
Obiettivi formativi	<p>The course aims at providing students with the ability to understand the principles of programming languages and the modern programming techniques, according to the main different paradigms:</p> <ul style="list-style-type: none"> • Basics of language theories and compilers • Object-oriented programming • Generic programming • Functional programming

The abilities to apply the listed knowledge elements regard mainly multi-paradigm development, fit for the context of distributed, heterogeneous and parallel systems:

- Handle correctly data and code in text form
- Acknowledge and use features shared by various languages
- Use static analysis to obtain robust and performant code
- Use polymorphism and metaprogramming to raise the level of abstraction
- Use functional programming for integrity and parallel computation

Prerequisiti

No propedeutic courses. However, students are expected to possess good basic abilities in structured and object-oriented programming.

Metodi didattici

Classroom lessons, using slides available to students in advance. Possibility to adopt the Flipped Classroom approach and seminars for some subjects. Programming exercises in laboratory.

Altre informazioni

Alternative and supplementary texts

- J. E. Hopcroft (2009). Automi, linguaggi e calcolabilità. 9788871925523.
- A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman (2009). Compilatori - Principi, tecniche e strumenti. 9788871925592.
- B. Stroustrup (2015). C++ - Linguaggio, libreria standard, principi di programmazione. 9788865184486.

Modalità di verifica dell'apprendimento

The exam consists of a written test, a programming test, which can also be performed during the lab exercises, and the presentation of a final project, which should apply multi-paradigm development to a real case.

Programma esteso

1. Introduction to formal languages (12 hours in classroom and 4 in lab)
 - a. Chomsky classification
 - b. Regular expressions
 - c. Context free grammars
 - d. Syntactic analysis and parser generators
 - e. Syntax trees and code generation
2. Object-oriented programming (4 hours in classroom and 2 in lab)
 - . Introduction
 - a. Object-oriented programming in C++14
 - b. Comparison with other languages: Java, Python, Go
3. Generic programming (4 hours in classroom and 2 in lab)
 - . Introduction
 - a. C++ templates and STL
4. Functional programming paradigm (8 hours in classroom and 4 in lab)
 - . Introduction
 - a. Presentation of LISP
 - b. Programming in Haskell
 - c. Functional features of C++14
 - d. Parallel programming in C++14
5. Dynamic languages and metaprogramming (4 hours in classroom and 2 in lab)
 - . Introduction
 - a. Dynamic features of C++14
 - b. Dynamic and scripting languages
 - c. Metaprogramming in Python