



| Principali informazioni sull'insegnamento | |
|---|---|
| Denominazione dell'insegnamento | Metodi Avanzati di Programmazione (corso B) / Advanced Programming Methods (class B) |
| Corso di studio | Laurea Triennale in Informatica |
| Anno di corso | Il anno |
| Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS): | 7+2 |
| SSD | ING-INF/05 |
| Lingua di erogazione | Italiano |
| Periodo di erogazione | Il semestre, 1 marzo 2022 – 4 giugno 2022 |
| Obbligo di frequenza | Non obbligatoria, ma fortemente consigliata. |

| Docente | |
|--|---|
| Nome e cognome | Pierpaolo Basile |
| Indirizzo mail | pierpaolo.basile@uniba.it |
| Telefono | +39 080 5442301 |
| Sede | CAMPUS Via E. Orabona, 4, Bari - Dipartimento di Informatica, VII piano, stanza 758 |
| Sede virtuale | MS Teams (fzo6f83) |
| Ricevimento (giorni, orari e modalità) | In presenza: Martedì 11:30-13:30. A distanza su piattaforma MS Teams previo prenotazione via mail. |

| Syllabus | |
|--|--|
| Obiettivi formativi | <ul style="list-style-type: none">• Progettare e implementare software (con particolare riferimento alla programmazione orientata agli oggetti)• Sviluppare modi efficaci ed efficienti per risolvere problemi con l'uso del computer |
| Prerequisiti | <ul style="list-style-type: none">• Programmazione imperativa• Algoritmi e strutture dati• Basi di dati |
| Contenuti di insegnamento (Programma) | <p>Introduzione al corso (1 ora)</p> <p>Introduzione ai paradigmi di programmazione: i tre approcci alla programmazione: operativa, definizionale e dimostrazionale (2 ore)</p> <p>L'astrazione nella programmazione (18 ore)</p> <ul style="list-style-type: none">• Fondamenti: Introduzione all'astrazione• Astrazione di funzione, di procedura, di controllo e di selettore• Astrazione di tipo e tipi astratti di dato• Specifiche algebriche e assiomatiche per i tipi astratti di dato• I moduli per l'incapsulamento dell'informazione e l'information hiding• Oggetti e classi di oggetti• Astrazione di dati: Tipo astratto di dato vs. classe di oggetti• Astrazione generica• Ambienti e linguaggi di programmazione <p>La programmazione orientata agli oggetti (35 ore)</p> |



| | |
|-----------------------------|--|
| | <ul style="list-style-type: none">• Fondamenti: oggetti, classi concrete, classi astratte, metaclassi, ereditarietà singola ed ereditarietà multipla, polimorfismo, gerarchia di classi e gerarchia di interfacce• Composizione di classi• Confronto tra ereditarietà e composizione nel riuso del software• Ambienti e linguaggi di programmazione• Java: caratteristiche generali del linguaggio• Java e Internet• Java vs. C++• Ambienti di sviluppo Java• Oggetti in Java: costruttori, distruttori, metodi, argomenti e valori di ritorno• Controllare il flusso di esecuzione: uso degli operatori Java, il controllo di esecuzione, l'inizializzazione• Nascondere le implementazioni: i package, i modificatori di accesso, le interfacce• Il riuso delle classi in Java: ereditarietà, derivazione, protetta, polimorfismo• I contenitori: array e collezioni• Approfondimenti su Java: il trattamento delle eccezioni, identificazione di tipo al run-time, programmazione generica in Java, il sistema I/O di Java• Connessione con le Basi di Dati: JDBC• Progettazione e creazione di interfacce per applicazioni: il package SWING• Programmazione in rete: server/client socket• Il multithreading: creazione di classi attive, sincronizzazione nell'accesso dei metodi• Estensione funzionale in Java: Lambda espressioni in Java, Pipeline e stream, Cenni di computazione in parallelo <p>Esercitazioni guidate in Java (30 ore)</p> <ul style="list-style-type: none">• Progetto di applicazioni con singole classi• Progetto di applicazioni con più classi organizzate gerarchicamente e in package• Progetto di applicazioni con classi astratte e uso del polimorfismo• Progetto di applicazioni con contenitori e trattamento delle eccezioni• Progetto di applicazioni con I/O da file• Progetto di connessione a database (JDBC)• Progetto di applicazioni con GUI mediante SWING• Progetto di applicazioni client-server e multithreading |
| Testi di riferimento | <p>A.L. Ambler, M.H. Burnett, & B.A. Zimmerman Operational Versus Definitional: A Perspective on Programming Paradigms IEEE Computer, 25(9): 28-43, September 1992.</p> <p>M. Shaw Abstraction Techniques in Modern Programming Languages IEEE Software, 10-26, October 1984.</p> <p>G. Masini, A. Napoli, D. Colnet, D. Léonard, & K. Tombre Linguaggi per la Programmazione a Oggetti (cap. 2-3, 6) Gruppo Editoriale Jackson, 1989.</p> <p>D. A. Watt Programming Language Concepts and Paradigms (cap. 5-6) Prentice Hall, 1990.</p> |



| | |
|-------------------------------------|---|
| | <p>Bruce Eckel Thinking in Java, 4th Edition (cap. 1-11, 13-14, 16-17, 19-20, 23-24) Prentice-Hall, 2006.</p> <p>Walter Savitch. Programmazione di base e avanzata con Java 2/ed Pearson Education, 2014.</p> <p>David J. Eck: Introduction to Programming Using Java, 7th ed.(v.7.0.2) 2015/16.</p> <p>J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley: The Java® Language Specification – Java SE 9 Edition. 2017.</p> |
| Note ai testi di riferimento | I testi di riferimento saranno integrati con slide e materiale didattico messo a disposizione dal docente sulla piattaforma di e-learning ADA. |

| | | | |
|---------------------------------------|--------------------|--|--------------------|
| Organizzazione della didattica | | | |
| Ore | | | |
| Totali | Didattica frontale | Pratica (laboratorio, campo, esercitazione, altro) | Studio individuale |
| 225 | 56 | 30 | 139 |
| CFU/ETCS | | | |
| 9 | 7 | 2 | |

| | |
|-------------------------|---|
| Metodi didattici | <p>Le lezioni frontali saranno dedicate all'apprendimento dei modelli teorici e dei concetti di base coadiuvati da alcuni esempi.</p> <p>Le ore di esercitazione saranno dedicate sia all'esecuzione di esercizi in classe coinvolgendo direttamente gli studenti nella risoluzione degli stessi, sia alla realizzazione di applicazioni di esempio in linguaggio Java.</p> <p>Si prevede l'utilizzo della piattaforma di e-learning del dipartimento (ADA) per la pubblicazione del materiale didattico, la discussione degli argomenti delle lezioni tra docente/studente e studenti/studenti, la condivisione dei risultati di laboratorio, la condivisione degli esercizi e la pubblicazione di materiale integrativo e di approfondimento.</p> <p>L'insegnamento non è erogato in modalità e-learning.</p> |
|-------------------------|---|

| | |
|--|---|
| Risultati di apprendimento previsti | |
| Conoscenza e capacità di comprensione | <ul style="list-style-type: none"> • Astrazione nella programmazione e nella definizione di strutture dati complesse • Progettazione e programmazione orientata agli oggetti |
| Conoscenza e capacità di comprensione applicate | <ul style="list-style-type: none"> • Saper definire le specifiche assiomatiche e algebriche di strutture dati complesse • Progettare software secondo il paradigma orientato agli oggetti • Saper sviluppare software utilizzando il linguaggio Java |
| Competenze trasversali | <ul style="list-style-type: none"> • <i>Autonomia di giudizio</i> <ul style="list-style-type: none"> ○ esercitazioni durante le lezioni ○ esercitazioni di laboratorio • <i>Abilità comunicative</i> <ul style="list-style-type: none"> ○ saper relazionare sulla progettazione e lo sviluppo di un'applicazione |



| | |
|--|---|
| | <p>sviluppata coerentemente con i principi del paradigma orientato agli oggetti</p> <ul style="list-style-type: none">• <i>Capacità di apprendere</i><ul style="list-style-type: none">○ orientarsi agilmente nelle problematiche che si presentano durante lo sviluppo di software progettato e realizzato coerentemente con i principi del paradigma orientato agli oggetti |
|--|---|

| Valutazione | |
|---|---|
| Modalità di verifica dell'apprendimento | Prova orale e caso di studio sviluppato in parte in laboratorio, realizzato in un gruppo di massimo tre studenti. |
| Criteri di valutazione | <ul style="list-style-type: none">• <i>Conoscenza e capacità di comprensione</i><ul style="list-style-type: none">○ saper definire strutture dati complesse○ saper progettare un software secondo i principi della programmazione orientata agli oggetti• <i>Conoscenza e capacità di comprensione applicate</i><ul style="list-style-type: none">○ saper definire le specifiche assiomatiche e algebriche di strutture dati complesse○ saper sviluppare un software utilizzando il linguaggio Java• <i>Autonomia di giudizio</i><ul style="list-style-type: none">○ svolgere le esercitazioni in classe○ svolgere le esercitazioni di laboratorio• <i>Abilità comunicative</i><ul style="list-style-type: none">○ presentare in maniera chiara e comprensibile il progetto realizzato come caso di studio• <i>Capacità di apprendere</i><ul style="list-style-type: none">○ dimostrare la capacità di sviluppare autonomamente nuove funzionalità nel progetto realizzato come caso di studio e nelle attività di laboratorio |
| Criteri di misurazione dell'apprendimento e di attribuzione del voto finale | <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 orale consiste nella presentazione del caso di studio e una serie di domande su tutti gli argomenti trattati durante l'insegnamento.</p> <p>La lode è assegnata agli studenti che dimostreranno ottime capacità di astrazione, ragionamento e creatività.</p> |
| Altro | |
| | |



| General information | |
|---|---|
| Academic subject | Advanced Programming Methods (class B) |
| Degree course | Bachelor's Degree in Computer Science |
| Academic Year | II year |
| CEuropean Credit Transfer and Accumulation System (ECTS): | 7+2 |
| Language | Italian |
| Academic calendar (starting and ending date) | II semester, 1st March 2022 – 4th June 2022 |
| Attendance | Not mandatory, but strongly recommended. |

| Professor / Lecturer | |
|-------------------------|---|
| Name and Surname | Pierpaolo Basile |
| E-mail | pierpaolo.basile@uniba.it |
| Telephone | +39 080 5442301 |
| Department and address | CAMPUS Via E. Orabona, 4, Bari - Dipartimento di Informatica, VII floor, room 758 |
| Virtual headquarters | MS Teams (fzo6f83) |
| Tutoring (time and day) | In-presence: Tuesday, 11:30-13:30. Remotely on MS Teams by booking via mail. |

| Syllabus | |
|----------------------|--|
| Learning Objectives | <ul style="list-style-type: none">• Design and develop software (with particular reference to object-oriented programming)• Develop effective and efficient methods to solve problems by computer |
| Course prerequisites | <ul style="list-style-type: none">• Imperative programming• Algorithms and data structures• Databases |
| Contents | Introduction (1 hour) Introduction to programming paradigms (2 hours) Abstraction in programming (18 hours) <ul style="list-style-type: none">• Introduction• Function/procedure abstraction• Control abstraction• Data abstraction• Algebraic and axiomatic specifications of data types• Information hiding• Objects and classes• Data abstraction: data type vs. class• Generic abstraction Object oriented programming <ul style="list-style-type: none">• Fundamentals: objects, classes, abstract classes, interfaces, metaclasses, single and multiple inheritance, polymorphism, class hierarchy• Composition• Inheritance vs. composition• Introduction to Java• Java and Internet |



| | |
|-------------------------------|--|
| | <ul style="list-style-type: none">• Java vs. C++• Java development environment• Objects in Java• Flow control in Java• Interfaces and packages• Inheritance and polymorphism in Java• Java data types and Collections• Java Exceptions• Java I/O• Java generics• JDBC• Java SWING• Sockets and threads in Java• Java Lambda expressions <p>Java laboratory (30 hours)</p> <ul style="list-style-type: none">• Projects with a single class• Projects with several classes (inheritance, composition, package)• Abstract classes and polymorphism• Java projects with Collections• Java projects with files and I/O• Java projects with JDBC• Java projects with SWING• Java projects with client/server sockets and threads |
| Books and bibliography | <p>A.L. Ambler, M.H. Burnett, & B.A. Zimmerman Operational Versus Definitional: A Perspective on Programming Paradigms IEEE Computer, 25(9): 28-43, September 1992.</p> <p>M. Shaw Abstraction Techniques in Modern Programming Languages IEEE Software, 10-26, October 1984.</p> <p>G. Masini, A. Napoli, D. Colnet, D. Léonard, & K. Tombre Linguaggi per la Programmazione a Oggetti (cap. 2-3, 6) Gruppo Editoriale Jackson, 1989.</p> <p>D. A. Watt Programming Language Concepts and Paradigms (cap. 5-6) Prentice Hall, 1990.</p> <p>Bruce Eckel Thinking in Java, 4th Edition (cap. 1-11, 13-14, 16-17, 19-20, 23-24) Prentice-Hall, 2006.</p> <p>Walter Savitch. Programmazione di base e avanzata con Java 2/ed Pearson Education, 2014.</p> <p>David J. Eck: Introduction to Programming Using Java, 7th ed. (v. 7.0.2) 2015/16.</p> <p>J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley: The Java® Language Specification – Java SE 9 Edition. 2017.</p> |
| Additional materials | The books will be integrated with slides and didactic material available on the ADA e-learning platform. |



| Work schedule | | | |
|---------------|----------|--|--|
| Hours | | | |
| Total | Lectures | Hands on (Laboratory, working groups, seminars, field trips) | Out-of-class study hours/ Self-study hours |
| 225 | 56 | 30 | 139 |
| ETCS | | | |
| 9 | 7 | 2 | |

| Teaching strategy |
|---|
| <p>Theoretical concepts will be provided during the lectures.</p> <p>During laboratory hours, exercises and Java applications will be executed in the classroom by directly involving students.</p> <p>The dissemination of the teaching material and the discussion between teacher/student and student/student will be supported by the ADA e-learning platform.</p> <p>The course is not delivered through e-learning.</p> |

| Expected learning outcomes |
|---|
| <p>Knowledge and understanding on:</p> <ul style="list-style-type: none"> Abstraction in programming and designing complex data structures Object oriented programming |
| <p>Applying knowledge and understanding on:</p> <ul style="list-style-type: none"> Define axiomatic and algebraic specifications of data types Design object oriented software Develop object oriented software |
| <p>Soft skills</p> <ul style="list-style-type: none"> <i>Making informed judgments and choices</i> <ul style="list-style-type: none"> exercises during lectures exercises during laboratory <i>Communicating knowledge and understanding</i> <ul style="list-style-type: none"> the student will be able to communicate concepts about object-oriented modelling and programming appropriately <i>Capacities to continue learning</i> <ul style="list-style-type: none"> deal with problems regarding the design and development of object-oriented software |

| Assessment and feedback |
|---|
| <p>Methods of assessment</p> <p>Oral exam and development of a case study (project) developed in part during the laboratory, carried out in a maximum of three students.</p> |
| <p>Evaluation criteria</p> <ul style="list-style-type: none"> <i>Knowledge and understanding</i> <ul style="list-style-type: none"> knowing how to define complex data structures knowing how to design object-oriented software <i>Applying knowledge and understanding</i> <ul style="list-style-type: none"> knowing how to define axiomatic and algebraic specifications of complex data types |



| | |
|---|--|
| | <ul style="list-style-type: none">• Java programming• <i>Autonomy of judgment</i><ul style="list-style-type: none">• carry out exercises during lectures• carry out exercises during laboratory• <i>Communication skills</i><ul style="list-style-type: none">• present clearly and understandably the project carried out as a case study• <i>Capacities to continue learning</i><ul style="list-style-type: none">• development of novel and innovative features in both the project and exercises |
| Criteria for assessment and attribution of the final mark | The final grade is on a 18-30 scale. During the oral exam, the student will present the project and answer questions about topics discussed in the course. 30 cum laude is assigned to students who demonstrate excellent abstraction, reasoning, and creativity skills. |
| Additional information | |
| | |