



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Metodi Avanzati di Programmazione (M-Z)	
Corso di studio	Corso di Laurea Triennale in Informatica	
Anno Accademico	2023/24	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	9 CFU	
Settore Scientifico Disciplinare	ING-INF 05	
Lingua di erogazione	Italiano	
Anno di corso	Secondo	
Periodo di erogazione	2^ semestre, le date esatte sono indicate annualmente nel manifesto/regolamento	
Obbligo di frequenza	La frequenza è fortemente raccomandata	
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-270/laurea-triennale-in-informatica-d.m.-270-1	

Docente/i	
Nome e cognome	Pierpaolo Basile
Indirizzo mail	pierpaolo.basile@uniba.it
Telefono	080-5442301
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.758, 7° piano
Sede virtuale	Piattaforma ADA - https://elearning.di.uniba.it
Sito web del docente	http://www.di.uniba.it/~swap/index.php?n=Membri.Basile
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Martedì 11:30-13:30, è preferibile prenotarsi via mail.

Syllabus	
Obiettivi formativi	Il corso si propone di introdurre i principi del paradigma Object Oriented consentendo allo studente di acquisire le competenze per sviluppare modi efficaci ed



	<p>efficienti per risolvere problemi con l'uso del computer mettendo a punto i metodi migliori per memorizzare ed accedere alle informazioni, rappresentarle, elaborarle e interpretarle programmando nel linguaggio Java.</p>
Prerequisiti	<p>Le seguenti conoscenze preliminari facilitano ed accelerano la comprensione degli argomenti dell'insegnamento:</p> <ul style="list-style-type: none">• Dall'insegnamento di Algoritmi e Strutture Dati: il linguaggio delle specifiche assiomatiche e le strutture dati;• Dall'insegnamento di Programmazione: le basi della programmazione imperativa, capacità di debugging, compilatori;• Dall'insegnamento di Architetture degli Elaboratori e SO: lo stack delle chiamate, la gestione della memoria, processi e thread;• Dall'insegnamento di Basi di dati: il linguaggio SQL.
Contenuti di insegnamento (Programma)	<ul style="list-style-type: none">• Introduzione al corso (1 ora)• Introduzione ai paradigmi di programmazione: i tre approcci alla programmazione: operativa, definizionale e dimostrazionale (2 ore)• L'astrazione nella programmazione (18 ore)<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• La programmazione orientata agli oggetti (35 ore)<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
<p>Testi di riferimento</p>	<ul style="list-style-type: none">• D. A. Watt Programming Language Concepts and Paradigms Prentice Hall, 1990. (capitoli 5,6 Astrazione nella programmazione, capitolo 12 Programmazione orientata agli oggetti)• Walter Savitch. Programmazione di base e avanzata con Java 2/ed Pearson Education, 2014 (Ambienti e linguaggi di programmazione, capitoli 1-16) <p>Gli studenti che lo desiderano possono ottenere i testi in prestito dalla Biblioteca. Può convenire verificarne la disponibilità mediante il Sistema Bibliotecario di Ateneo https://opac.uniba.it/easyweb/w8018/index.php? e contattare la biblioteca per concordare il prestito.</p> <p>Testi e articoli scientifici aggiuntivi (per approfondimenti)</p> <ul style="list-style-type: none">• Bruce Eckel Thinking in Java, 4th Edition Prentice-Hall, 2006 (capitoli 1-20, 22 per la parte relativa ad Ambienti e linguaggi di programmazione)• 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 (Introduzione ai paradigmi di programmazione)• M. Shaw Abstraction Techniques in Modern Programming Languages IEEE Software, 10-26, October 1984 (Astrazione nella programmazione)• G. Masini, A. Napoli, D. Colnet, D. Léonard, & K. Tombre Linguaggi per la Programmazione a Oggetti Gruppo Editoriale Jackson, 1991 (capitoli 2-3, 6 Programmazione orientata agli oggetti)• https://docs.oracle.com/javase/specs/ (per ultimi aggiornamenti su Java) <p>Gli studenti che lo desiderano possono ottenere i testi in prestito dalla Biblioteca. Può convenire verificarne la disponibilità mediante il Sistema Bibliotecario di Ateneo https://opac.uniba.it/easyweb/w8018/index.php? e contattare la biblioteca per concordare il prestito.</p>
<p>Note ai testi di riferimento</p>	<p>Nel corso delle lezioni il docente illustrerà i concetti con l'ausilio di slide che sintetizzano i contenuti del testo di riferimento e riportano esercizi svolti in aula. Le slide saranno rese disponibili al termine di ogni lezione sulla piattaforma ADA del dipartimento.</p> <p>Sulla piattaforma ADA sono disponibili:</p> <ul style="list-style-type: none">• slide di supporto utilizzate dal docente durante le lezioni;• esercizi con soluzioni;• materiale integrativo fornito dal docente. <p>La traccia del caso di studio sarà fornita durante le lezioni di laboratorio.</p>



Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (laboratorio, esercitazione)	Studio individuale
86 ore	56 ore	30 ore	139 ore
CFU/ETCS			
9 CFU	7 CFU	2 CFU	

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>

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	<ul style="list-style-type: none">• Capacità di astrazione nella programmazione e nella definizione di strutture dati complesse• Conoscenza del paradigma di programmazione orientata agli oggetti• Capacità di 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	<p>Autonomia di giudizio Lo studente acquisterà autonomia di giudizio rispetto alla realizzazione in Java di software robusti</p> <p>Capacità di apprendere in modo autonomo Lo studente svilupperà capacità di apprendere e di orientarsi agilmente nelle problematiche che si presentano durante lo sviluppo di software realizzato in Java coerentemente con i principi del paradigma OO.</p> <p>Abilità comunicative</p>



	Lo studente saprà relazionarsi in maniera appropriata in riferimento ai principi della modellazione e programmazione orientata ad oggetti e Java
--	--

Valutazione	
Modalità di verifica dell'apprendimento	<ul style="list-style-type: none">• Caso di studio sviluppato in parte durante le ore di esercitazione/laboratorio e in parte durante le ore di studio individuale previsti dai crediti di esercitazione/laboratorio. La consegna deve avvenire 5 giorni prima della data dell'appello e deve includere: relazione e diagramma delle classi, java doc, guida utente, codice, eseguibile. La traccia è assegnata durante le lezioni di laboratorio (e contestualmente pubblicata su ADA). Lo svolgimento del caso di studio può essere in gruppi di massimo 3 componenti.• Prova orale: in aula, viene discusso il caso di studio e verificata, tramite domande, la conoscenza degli argomenti trattati durante il corso; votazione massima 30/30.
Criteria di valutazione	<ul style="list-style-type: none">• Conoscenza e capacità di comprensione:<ul style="list-style-type: none">○ Capacità di comprendere le domande formulate per la prova orale e rispondere in maniera pertinente ed esaustiva○ Capacità di comprendere le linee guida per lo svolgimento delle attività di laboratorio e del caso di studio• Conoscenza e capacità di comprensione applicate:<ul style="list-style-type: none">○ Conoscenza esaustiva degli argomenti oggetto del corso e loro utilizzo nello svolgimento del caso di studio, trattazione delle questioni teoriche• Autonomia di giudizio:<ul style="list-style-type: none">○ Saper verificare se il caso di studio realizzato rispetta le buone prassi della programmazione ad oggetti e dell'astrazione• Abilità comunicative:<ul style="list-style-type: none">○ Capacità di rispondere ai quesiti formulati per la prova orale in maniera corretta, esaustiva e utilizzando appropriatamente il linguaggio tecnico• Capacità di apprendere:<ul style="list-style-type: none">○ Comprensione dei contenuti del corso e capacità di utilizzare i concetti appresi nello svolgimento di esercizi e nello sviluppo del caso di studio in Java
Criteria 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>Durante la prova orale verranno presi in considerazione differenti criteri:</p> <ul style="list-style-type: none">• Qualità complessiva del caso di studio svolto• Capacità di applicare i principi della programmazione ad oggetti e dell'astrazione in modo corretto• Applicazione all'interno del caso di studio dei vari argomenti trattati durante il laboratorio• Documentazione e presentazione del caso di studio• Esattezza e chiarezza delle risposte alle domande durante la prova orale <p>La lode è assegnata agli studenti che dimostreranno ottime capacità di astrazione, ragionamento e creatività.</p>



Altro

Si suggerisce agli studenti di affidarsi esclusivamente alle informazioni/comunicazioni fornite sui siti ufficiali del Dipartimento di Informatica, ovvero sui gruppi social solo se costituiti e amministrati esclusivamente dai docenti dei relativi insegnamenti:

- <https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea>
- <https://www.uniba.it/it/ricerca/dipartimenti/informatica>
- <https://elearning.uniba.it/>

I programmi degli insegnamenti sono disponibili qui:

- <https://elearning.uniba.it/>

Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei Regolamenti didattici e manifesti degli studi disponibili nel sito:

- <https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea>

Si suggerisce agli studenti di diffidare delle informazioni e dei materiali circolanti su siti o gruppi social non ufficiali, poiché spesso sono risultati non affidabili, non corretti o incompleti. Per ogni dubbio, chiedere un incontro al docente secondo le modalità previste per il ricevimento.

-
- link al corso sulla piattaforma e-learning UNIBA:
<https://elearning.uniba.it/course/view.php?id=1977>
 - codice MS TEAMS: **5nvego4**



Main information on the course

Course name	Advanced Programming Methods (M-Z)	
Degree	Bachelor's Degree in Computer Science	
Academic year	2023/24	
European Credit Transfer and Accumulation System (ECTS), in Italian Crediti Formativi Universitari (CFU)	7 CFU T1 + 2 CFU T2 (each CFU corresponds to 25 hours (h) of student's time); CFU are of type T1, T2 or T3 T1 = 8 h lecture + 17 h individual study T2 = 15 h practice + 10 h individual study T3 = 25 h individual study	
Settore Scientifico Disciplinare	ING-INF 05	
Course language	Italian	
Anno di corso	Third	
Periodo di erogazione	2nd semester, the exact dates are reported in the regulation of the specific academic year	
Obbligo di frequenza	It is highly recommended to attend classes	
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-270/laurea-triennale-in-informatica-d.m.-270-1	

Teacher(s)	
Name and Surname	Pierpaolo Basile
email	pierpaolo.basile@uniba.it
phone	080-5442301
office	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.758, 7° floor
e-learning platform	ADA Website - https://elearning.di.uniba.it
Teacher's homepage	http://www.di.uniba.it/~swap/index.php?n=Membri.Basile
Office hours	Tuesdays 11:30 a.m. to 1:30 p.m. Reservations via email are preferred.

Syllabus	
Course goals	The course aims to introduce the principles of the Object Oriented paradigm by enabling the student to acquire the skills to develop effective and efficient ways to solve problems using computers by devising the best methods for storing and accessing, representing, processing and interpreting information by using the Java language.
Prerequisites/requirements	The following background knowledge facilitates and accelerates understanding of the teaching topics: <ul style="list-style-type: none">● from Algorithms and Data Structures: abstraction and data structures;● from Programming: basics of imperative programming;● from Operating Systems and Computers Architecture: calls stack, memory management, processes and threads;● from Databases: SQL.
Course program	<ul style="list-style-type: none">● Introduction (1 ora)● Introduction to programming paradigms (2 ore)● Abstraction in programming (18 ore)



	<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 (35 ore)<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○ 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○ Java RESTful <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 <p>Java projects with client/server sockets and threads</p>
<p>Books of reference</p>	<p>D. A. Watt Programming Language Concepts and Paradigms Prentice Hall, 1990. (<i>capitoli 5,6 Astrazione nella programmazione, capitolo 12 Programmazione orientata agli oggetti</i>)</p> <p>Walter Savitch. Programmazione di base e avanzata con Java 2/ed Pearson Education, 2014 (<i>Ambienti e linguaggi di programmazione, capitoli 1-16</i>)</p> <p>Students can obtain the texts on loan from the library. It is suggested that students check whether the texts are available through the University Library System https://opac.uniba.it/easyweb/w8018/index.php? and contact the library.</p> <p>Additional scientific texts and articles (for further study)</p> <p>Bruce Eckel Thinking in Java, 4th Edition Prentice-Hall, 2006 (<i>capitoli 1-20, 22 per la parte relativa ad Ambienti e linguaggi di programmazione</i>)</p>



	<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 (<i>Introduzione ai paradigmi di programmazione</i>)</p> <p>M. Shaw Abstraction Techniques in Modern Programming Languages IEEE Software, 10-26, October 1984 (<i>Astrazione nella programmazione</i>)</p> <p>G. Masini, A. Napoli, D. Colnet, D. Léonard, & K. Tombre Linguaggi per la Programmazione a Oggetti Gruppo Editoriale Jackson, 1991 (<i>capitoli 2-3, 6 Programmazione orientata agli oggetti</i>)</p> <p>https://docs.oracle.com/javase/specs/ (for the latestest Java updates)</p>		
<p>Notes to the books</p>	<p>The lecturer will illustrate the concepts with the help of slides that summarize the contents of the reference text and report exercises carried out in the classroom. The slides will be made available at the end of each lecture on the ADA website.</p> <p>Available on the ADA website are:</p> <ul style="list-style-type: none"> ● supporting slides used during lectures; ● exercises with solutions; ● supplementary materials provided by the lecturer. <p>The case study instructions will be provided during laboratory lectures.</p>		
<p>Organization of the didactic activities</p>			
<p>Hours</p>			
<p>Total</p>	<p>Lectures</p>	<p>Practice sessions</p>	<p>Individual study</p>
<p>86 hours</p>	<p>56 hours</p>	<p>30 hours</p>	<p>139 hours</p>
<p>CFU/ETCS</p>			
<p>9 CFU</p>	<p>7 CFU T1</p>	<p>2 CFU T2</p>	
<p>Teaching methods</p>			
	<p>Lectures will be devoted to learning the theoretical models and basic concepts aided by some examples.</p> <p>Exercise hours will be devoted both to making in-class exercises directly involving students in solving them and to the implementation of example applications.</p> <p>It is planned to use the department's e-learning platform (ADA) for posting of lecture material, discussion of lecture topics between lecturer/student and student/students, sharing of lab results, sharing of exercises, and posting of supplementary and in-depth material.</p>		
<p>Expected learning outcomes</p>			
<p>Knowledge and understanding</p>	<ul style="list-style-type: none"> ● Abstraction in programming and designing complex data structures ● Object-oriented programming 		



<p>Applying knowledge and understanding</p>	<ul style="list-style-type: none"> ● Define axiomatic and algebraic specifications of data types ● Design object-oriented software ● Develop object-oriented software
<p>Other skills</p>	<p>Autonomy of judgment The student will gain autonomy with respect to Java implementation of robust software.</p> <p>Communication skills The student will be able to communicate appropriately the principles of object-oriented modelling and programming and Java</p> <p>Ability to learn independently The student will develop abilities to learn and manage issues that arise during the development of software implemented in Java consistent with the principles of the OO paradigm.</p>

Assessment	
<p>Assessment methods</p>	<ul style="list-style-type: none"> ● Case study developed partly during lab hours and partly during individual study hours. Submission is due five days prior to the exam date and must include: a report and class diagram, java doc, user guide, code, and executable. The outline is assigned during lab classes (and contextually posted on ADA). The case study can be developed in groups of up to 3 components. ● Oral test: in the classroom, the case study is discussed and knowledge of the topics covered in the course is evaluated through questions; maximum grade 30/30.
<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 ○ 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



Measurements and final grade	<p><u>Final Exam</u></p> <p>The evaluation of the exam is given in thirtieths. During the exam, the student will present the case study and answers questions on all topics covered during the teaching. Different criteria will be considered during the evaluation:</p> <ul style="list-style-type: none">● Overall quality of the case study● Ability to apply the principles of object-oriented programming and abstraction properly● Application within the case study of the various topics covered during the course● Documentation and presentation of the case study● Accuracy and clarity of answers to questions during the oral test <p>The test is passed with a grade greater than or equal to 18. Honors are awarded to students who demonstrate excellent abstraction, reasoning and creativity skills.</p>
Further information	<p>It is suggested that students rely exclusively on the information/communication provided on the official websites of the Department of Computer Science, or on social groups only if they are formed and administered exclusively by the faculty members:</p> <ul style="list-style-type: none">● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea● https://www.uniba.it/it/ricerca/dipartimenti/informatica● https://elearning.uniba.it/ <p>Teaching programs are available here:</p> <ul style="list-style-type: none">● https://elearning.uniba.it/ <p>Information that all students should know is written in the Teaching Regulations available on the website:</p> <ul style="list-style-type: none">● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea <p>Students are suggested to be wary of information and materials circulating on unofficial sites or social groups, as they are often found to be unreliable, incorrect or incomplete. If you have any doubts, please schedule a meeting with the lecturer in accordance with the reception arrangements.</p> <ul style="list-style-type: none">● link to the course on the UNIBA e-learning website: https://elearning.uniba.it/course/view.php?id=1977● MS TEAMS code: 5nvego4