



## Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	<b>Metodi Avanzati di Programmazione (A-L)</b>	
Corso di studio	Corso di Laurea Triennale in Informatica	
Anno Accademico	2024/25	
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	<a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-270/laurea-triennale-in-informatica-d.m.-270-1">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-270/laurea-triennale-in-informatica-d.m.-270-1</a>	

<b>Docente/i</b>	
Nome e cognome	Annalisa Appice
Indirizzo mail	<a href="mailto:annalisa.appice@uniba.it">annalisa.appice@uniba.it</a>
Telefono	080544 3262
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n. 512, V piano.
Sede virtuale	Piattaforma e-learning- <a href="https://elearning.uniba.it/">https://elearning.uniba.it/</a>
Sito web del docente	<a href="http://www.di.uniba.it/~appice/">http://www.di.uniba.it/~appice/</a>
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Martedì 15:00-17:00 (da confermare su appuntamento)

## Syllabus



<b>Obiettivi formativi</b>	Il corso si propone di introdurre i principi del paradigma Object Oriented consentendo allo studente di acquisire le competenze per sviluppare modi efficaci ed 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
<b>Prerequisiti</b>	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; Dall'insegnamento di Matematica discreta: il principio di induzione, il concetto di algebra omogenea ed eterogenea.
<b>Contenuti di insegnamento (Programma)</b>	<b>Introduzione ai paradigmi di programmazione (2 ore)</b> <ul style="list-style-type: none"><li>● Tre approcci alla programmazione: operativa, definizionale e dimostrazionale.</li></ul> <b>Astrazione nella programmazione (10 ore)</b> <ul style="list-style-type: none"><li>● Fondamenti: Introduzione all'astrazione. Astrazione di funzione, di procedura, di controllo, e di selettore. Astrazione di tipo e tipi astratti di dato. Specifiche algebriche 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. I moduli in Ada.</li></ul> <b>Programmazione orientata agli oggetti (14 ore)</b> <ul style="list-style-type: none"><li>● Fondamenti: oggetti, classi concrete, classi astratte, 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.</li></ul> <b>Ambienti e linguaggi di programmazione (30 ore)</b> <ul style="list-style-type: none"><li>● Oggetti in Java: costruttori; metodi.</li><li>● Controllare il flusso di esecuzione: uso degli operatori Java; il controllo di esecuzione; l'inizializzazione.</li><li>● Nascondere le implementazioni: i package; i modificatori di accesso; le interfacce.</li><li>● Il riuso delle classi in Java: ereditarietà, derivazione protetta; polimorfismo. I contenitori: array; collezioni; insiemi.</li><li>● Approfondimenti su Java: il trattamento delle eccezioni; identificazione di tipo al run-time; programmazione generica in Java; il sistema I/O di Java.</li><li>● Connessione con le Basi di Dati: JDBC.</li><li>● Programmazione in rete: socket. Il multithreading: creazione di classi attive; sincronizzazione nell'accesso dei metodi.</li><li>● Estensione funzionale in Java: Lambda espressione in Java, Pipeline e stream, Cenni di computazione in parallelo</li></ul> <b>Esercitazioni guidate e laboratorio (30 ore)</b> <ul style="list-style-type: none"><li>● Astrazione delle strutture dati di un software con il linguaggio delle specifiche algebriche</li><li>● Java: caratteristiche generali del linguaggio; Java e Internet; Java vs. C++. Ambienti di sviluppo Java. L'ambiente Eclipse.</li><li>● Esercizi e progetto di applicazioni con singole classi; applicazioni con più classi organizzate gerarchicamente e in package; applicazioni con classi astratte e uso del polimorfismo; applicazioni con contenitori e trattamento delle eccezioni; applicazioni con I/O da file; connessione a database; di applicazioni client-server e multithreading.</li></ul>



<p><b>Testi di riferimento</b></p>	<p>D. A. Watt Programming Language Concepts and Paradigms Prentice Hall, 1990. (<b>capitoli 5,6 Astrazione nella programmazione, capitolo 12 Programmazione orientata agli oggetti</b>)</p> <p>Walter Savitch. Programmazione di base e avanzata con Java 2/ed Pearson Education, 2014 (<b>Ambienti e linguaggi di programmazione, capitoli 1-16</b>)</p> <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 <a href="https://opac.uniba.it/easyweb/w8018/index.php?">https://opac.uniba.it/easyweb/w8018/index.php?</a> e contattare la biblioteca per concordare il prestito.</p> <p>Testi e articoli scientifici aggiuntivi (per approfondimenti)</p> <p>Bruce Eckel Thinking in Java, 4th Edition Prentice-Hall, 2006 (<b>capitoli 1-20, 22</b> per la parte relativa ad <b>Ambienti e linguaggi di programmazione</b> )</p> <p>A.L. Ambler, M.H. Burnett, &amp; B.A. Zimmerman Operational Versus Definitional: A Perspective on Programming Paradigms IEEE Computer, 25(9): 28-43, September 1992 (<b>Introduzione ai paradigmi di programmazione</b>)</p> <p>M. Shaw Abstraction Techniques in Modern Programming Languages IEEE Software, 10-26, October 1984 (<b>Astrazione nella programmazione</b>)</p> <p>G. Masini, A. Napoli, D. Colnet, D. Léonard, &amp; K. Tombre Linguaggi per la Programmazione a Oggetti Gruppo Editoriale Jackson, 1991 (<b>capitoli 2-3, 6 Programmazione orientata agli oggetti</b>)</p> <p><a href="https://docs.oracle.com/javase/specs/">https://docs.oracle.com/javase/specs/</a> (per ultimi aggiornamenti su Java)</p>		
<p><b>Note ai testi di riferimento</b></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 (v. sopra 'sede virtuale').</p> <p>Sulla piattaforma ADA sono disponibili:</p> <ul style="list-style-type: none"> <li>● slide di supporto utilizzate dal docente durante le lezioni;</li> <li>● esercizi con soluzioni;</li> <li>● alcune tracce di prove scritte di esami precedenti.</li> </ul> <p>La traccia del progetto sarà fornita durante le lezioni di laboratorio che inizieranno approssimativamente a fine marzo 2025</p>		
<p><b>Organizzazione della didattica</b></p>			
<p><b>Ore</b></p>			
<p>Totali</p>	<p>Didattica frontale</p>	<p>Pratica (laboratorio, progetto, esercitazione, altro)</p>	<p>Studio individuale</p>
<p>86 ore</p>	<p>56 ore</p>	<p>30 ore</p>	<p>139 ore</p>



CFU/ETCS			
9 CFU	7 CFU	2 CFU	

Metodi didattici	
	<p>Didattica in aula con lezioni di teoria, esercitazioni guidate durante le quali gli studenti interagiranno con il docente per lo svolgimento degli esercizi, attività di laboratorio per la programmazione e svolgimento di progetto in gruppo. I gruppi possono includere un numero di componenti che varia da 1 a 3. L'organizzazione in gruppi è decisa liberamente dagli studenti, ma va comunicata al docente al momento della consegna del progetto e non può essere successivamente modificata.</p> <p>L'attività di esercitazione, laboratorio, e progetto sono finalizzate ad affiancare pratica e teoria per lo sviluppo di software e soluzione di problemi con l'uso del computer.</p>

Risultati di apprendimento previsti	
<b>Conoscenza e capacità di comprensione</b>	La classe acquisirà una conoscenza di base della modellazione e programmazione Orientata agli Oggetti (OO).
<b>Conoscenza e capacità di comprensione applicate</b>	Attraverso l'introduzione del linguaggio di programmazione Java e lo sviluppo guidato di un progetto la classe approfondirà la programmazione orientata a oggetti, la composizione di classi, l'uso di gerarchie di classi ed alcune strutture dati fondamentali, lo sviluppo di applicazioni client server, la programmazione funzionale in Java
<b>Competenze trasversali</b>	<p><b>Autonomia di giudizio</b> La classe acquisterà autonomia di giudizio rispetto alla realizzazione in Java di software robusti</p> <p><b>Capacità di apprendere in modo autonomo</b> La classe 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><b>Abilità comunicative</b> La classe relazionerà in maniera appropriata in riferimento ai principi della modellazione e programmazione orientata ad oggetti e Java</p>



Valutazione	
Modalità di verifica dell'apprendimento	<ul style="list-style-type: none"><li>● Prova scritta: in aula, 3 domande aperte su teoria ed esercizi in merito ad argomenti del syllabo; tempo assegnato 90 minuti; votazione massima 33/33. La prova scritta si ritiene superata se lo studente consegue una votazione maggiore uguale di 18/33.</li><li>● Progetto: la consegna deve avvenire con weTransfer indirizzato al docente di riferimento e deve includere diagramma delle classi, java doc, casi di test, guida utente, codice, eseguibile e versione estesa (opzionale). La traccia è assegnata durante le lezioni di laboratorio (e contestualmente pubblicata su ADA). Lo svolgimento del progetto può essere in gruppi di massimo 3 componenti. La votazione massima assegnata al progetto è 33/30. Il progetto è rigettato se non è completo in tutte le sue parti. La prova di progetto si ritiene superata se gli viene assegnata una votazione maggiore uguale di 18/33.</li><li>● Il progetto assegnato è valido solo per gli appelli erogati nell'AA 2024-25</li></ul> <p>Voto finale è la media del voto conseguito nella prova scritta e progetto arrotondato per eccesso. Un voto maggiore di 30 comporta l'assegnazione della lode</p>
Criteri di valutazione	<ul style="list-style-type: none"><li>● Conoscenza e capacità di comprensione:<ul style="list-style-type: none"><li>○ Capacità di comprendere le domande formulate per la prova scritta e rispondere in maniera pertinente ed esaustiva</li><li>○ Capacità di comprendere le linee guida per lo svolgimento delle attività di laboratorio e del progetto</li></ul></li><li>● Conoscenza e capacità di comprensione applicate:<ul style="list-style-type: none"><li>○ Conoscenza esaustiva degli argomenti oggetto del corso e loro utilizzo nello svolgimento di esercizi oggetto della prova scritta, trattazione delle questioni teoriche e realizzazione del progetto</li></ul></li><li>● Autonomia di giudizio:<ul style="list-style-type: none"><li>○ La prova scritta include tre domande di teoria e/o esercizi su temi inclusi nel programma del corso. Ciascuna domanda è valutata al massimo 11 punti (sulla base della completezza e correttezza della risposta). La prova si ritiene superata se lo studente consegue almeno 18/33 nello svolgimento dell'intera prova.</li><li>○ Il progetto è sviluppato seguendo le linee guida fornite durante le lezioni di laboratorio. Il progetto deve includere una breve descrizione che enuclei le funzionalità implementate e i punti di differenza rispetto al progetto originale, diagramma delle classi/package UML (max 2 punti), JAVADOC (max 2 punti), guida utente del sistema e casi di test (max 2 punti), jar + script per esecuzione del jar (max 2 punti), codice sorgente (15 punti), estensioni (10 punti). Il progetto è ritenuto sufficiente se valutato almeno 18/33</li></ul></li><li>● Abilità comunicative:<ul style="list-style-type: none"><li>○ Capacità di rispondere ai quesiti formulati per la prova scritta in maniera corretta, esaustiva e utilizzando appropriatamente il linguaggio tecnico</li></ul></li><li>● Capacità di apprendere:</li></ul> <p>Comprensione dei contenuti del corso e capacità utilizzare i concetti appresi nello svolgimento di esercizi e sviluppo del progetto di software in Java</p>
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	<p>Il voto finale è la media aritmetica dei voti conseguiti nella prova scritta e progetto arrotondato per eccesso.</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/corsi-di-laurea>
- <https://www.uniba.it/it/ricerca/dipartimenti/informatica>
- <https://elearning.di.uniba.it/>

I programmi degli insegnamenti sono disponibili qui:

- <https://programmi.di.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 dalle informazioni circolanti su siti o gruppi social non ufficiali, poichè spesso sono risultate non affidabili, non corrette o incomplete.

---

Gli studenti potranno unirsi al forum del corso A.A. 2024/25 iscrivendosi al corso sulla piattaforma e-learning del dipartimento ADA: <https://elearning.di.uniba.it/>



## Main information on the course

Course name	<b>Advanced Methods for Programming</b>	
Bachelor degree	Bachelor degree in Informatics	
Academic Year	2024/245	
University credits (CFU) / European Credit Transfer and Accumulation System (ECTS)	9 CFU	
SSD	ING-INF 05	
Language	Italian	
Year	Second	
Time	2 <sup>^</sup> semester, the exact dates are shown in the poster/regulations	
Attendance	Attendance is strongly recommended	
Web site	<a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-270/laurea-triennale-in-informatica-d.m.-270-1">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-270/laurea-triennale-in-informatica-d.m.-270-1</a>	

## Teacher

Name and surname	Annalisa Appice
Email	<a href="mailto:annalisa.appice@uniba.it">annalisa.appice@uniba.it</a>
Phone	080544 3262
Location	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n. 512, V piano.
Virtual location	Piattaforma e-learning- <a href="https://elearning.uniba.it/">https://elearning.uniba.it/</a>
Teacher Web site	<a href="http://www.di.uniba.it/~appice/">http://www.di.uniba.it/~appice/</a>
Meetings	Tuesday 15:00-17:00 (on appointment)

## Syllabus

<b>Training objectives</b>	The course aims to introduce the principles of the Object Oriented paradigm allowing the student to acquire the skills to develop effective and efficient ways to solve problems with the use of computers by developing the best methods for storing and accessing information, representing it and processing it and interpreting them by programming in the Java language
<b>Prerequisites</b>	From the course of Algorithms and Data Structures: the language of axiomatic specifications and data structures; From the course of Programming: the basics of imperative programming, debugging skills, compilers; From the course of Computer Architectures and OS: the call stack, memory management, processes and threads; From the course of Databases: the SQL language; From the course of Discrete Mathematics: the principle of induction, the concept of homogeneous and heterogeneous algebra.
<b>Programme</b>	Introduction to programming paradigms (2 hours) • Three approaches to programming: operational, definitional and demonstration.  Abstraction in programming (10 hours)



	<ul style="list-style-type: none"> <li>● Fundamentals: Introduction to abstraction. Abstraction of function, procedure, control, and selector. Type abstraction and abstract data types. Algebraic specifications for abstract data types. Modules for information encapsulation and information hiding. Objects and object classes. Data Abstraction: Abstract Data Type vs. class of objects. Generic abstraction. The modules in Ada.</li> </ul> <p>Object-Oriented Programming (14 hours)</p> <ul style="list-style-type: none"> <li>● Fundamentals: objects, concrete classes, abstract classes, single inheritance and multiple inheritance, polymorphism, class hierarchy and interface hierarchy. Composition of classes. Comparing inheritance and composition in software reuse.</li> </ul> <p>Programming environments and languages (30 hours)</p> <ul style="list-style-type: none"> <li>● Objects in Java: constructors; methods.</li> <li>● Control the execution flow: use of Java operators; execution control; initialization.</li> <li>● Hide implementations: packages; access modifiers; the interfaces.</li> <li>● Reuse of classes in Java: inheritance, protected derivation; polymorphism. Containers: array; collections; sets.</li> <li>● Insights into Java: exception handling; type identification at run-time; generic programming in Java; the Java I/O system.</li> <li>● Connection with Databases: JDBC.</li> <li>● Network programming: sockets. Multithreading: creation of active classes; synchronization in method access.</li> <li>● Functional extension in Java: Lambda expression in Java, Pipelines and streams, Notes on parallel computation</li> </ul> <p>Guided exercises and laboratory (30 hours)</p> <ul style="list-style-type: none"> <li>● Abstraction of software data structures with the algebraic specification language</li> <li>● Java: general characteristics of the language; Java and the Internet; Java vs. C++. Java development environments. The Eclipse environment.</li> <li>● Exercises and application design with single classes; applications with multiple classes organized hierarchically and in packages; applications with abstract classes and use of polymorphism; applications with containers and exception handling; applications with file I/O; database connection; of client-server and multithreading applications.</li> </ul>
<p><b>Books</b></p>	<p>D. A. Watt Programming Language Concepts and Paradigms Prentice Hall, 1990. (chapters 5,6 Abstraction in programming, chapter 12 Object-oriented programming)</p> <p>Walter Savitch. Basic and Advanced Programming with Java 2/ed Pearson Education, 2014 (Programming Languages and Environments, Chapters 1-16)</p> <p>Students who wish can borrow texts from the Library. Is it convenient to check availability via the University Library System <a href="https://opac.uniba.it/easyweb/w8018/index.php?">https://opac.uniba.it/easyweb/w8018/index.php?</a> and contact the library to arrange the loan.</p> <p>Additional scientific texts and articles (for further information)</p> <p>Bruce Eckel Thinking in Java, 4th Edition Prentice-Hall, 2006 (chapters 1-20, 22 for the part relating to Programming Environments and Languages)</p> <p>TO THE. Ambler, M.H. Burnett, &amp; B.A. Zimmerman Operational Versus Definitional: A Perspective on Programming Paradigms IEEE Computer, 25(9): 28-43, September 1992 (Introduction to Programming Paradigms)</p> <p>M. Shaw Abstraction Techniques in Modern Programming Languages IEEE Software, 10-26, October 1984 (Abstraction in programming)</p>



	<p>G. Masini, A. Napoli, D. Colnet, D. Léonard, &amp; K. Tombre Languages for Object-Oriented Programming Jackson Publishing Group, 1991 (chapters 2-3, 6 Object-Oriented Programming)  <a href="https://docs.oracle.com/javase/specs/">https://docs.oracle.com/javase/specs/</a> (for latest updates on Java)</p>		
<b>Notes to books</b>	<p>During the lessons, the teacher will illustrate the concepts with the help of slides that summarize (and sometimes integrate) 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 lesson on the department's ADA platform (see 'virtual venue' above).</p> <p>The following are available on the ADA platform:</p> <ul style="list-style-type: none"> <li>• support slides used by the teacher during lessons;</li> <li>• exercises with solutions;</li> <li>• some traces of written tests from previous exams.</li> </ul> <p>The project outline will be provided during the laboratory lessons starting approximately at the end of October March 2025 (the outline will be published on ADA).</p>		
<b>Organization of lectures</b>			
<b>Hours</b>			
Total	Frontal teaching	Practice (laboratory, project, exercise, other)	Individual study
86 hours	56 hours	30 hours	139 hours
<b>CFU/ETCS</b>			
9 CFU	7 CFU	2 CFU	
<b>Teaching Methods</b>			
	<p>Classroom teaching with theory lessons, guided exercises during which students will interact with the teacher to carry out the exercises, laboratory activities for planning and carrying out the project (possibly in a group). The groups can include a number of members ranging from 1 to 3. The organization into groups is freely decided by the students, but must be communicated to the teacher at the time of delivery of the project and cannot be subsequently modified.</p> <p>The exercise, laboratory and project activities are aimed at combining practice and theory for the development of software and problem solving with the use of computers</p>		
<b>Expected Learning Results</b>			
<b>Knowledge and Understanding</b>	<p>The class will acquire a basic understanding of Object Oriented (OO) modeling and programming.</p>		
<b>Applied Knowledge and Understanding</b>	<p>Through the introduction of the Java programming language and the guided development of a project, the class will delve into object-oriented programming, class composition, the use of class hierarchies and some fundamental data structures, the development of client server applications, functional programming in Java</p>		
<b>Transversal skills</b>	<p>Autonomy of judgement  The class will acquire autonomy of judgment regarding the creation of robust software in Java</p> <p>Ability to learn independently</p>		



	<p>The class will develop the ability to learn and easily orient itself in the problems that arise during the development of software created in Java consistently with the principles of the OO paradigm.</p> <p>Communication skills The class will relate appropriately to the principles of object-oriented and Java modeling and programming</p>
--	---

Evaluation	
<b>Learning assessment methods</b>	<p>Written test + Project</p> <ul style="list-style-type: none"> <li>● Written test in the classroom, 3 open questions on theory and exercises regarding the topics of the syllabus; allotted time 90 minutes; maximum score 33/33. The written test is considered passed if the student achieves a grade greater than 18/33. The use of a traditional calculator is permitted during the test (no PCs, smartphones, tablets).</li> <li>● Project: delivery must take place with weTransfer addressed to the institutional email of the course reference teacher within 24 hours before the discussion of the project and must include the code produced in Python and the power point presentation of the project activity carried out. During the discussion of the project, the student has 10 minutes to present the project with Power Point slides and is then asked to answer the teacher's question regarding the way in which the code corresponding to the development of the project was written.</li> </ul> <p>The project code can be developed in a group (maximum 3 members), but the power-point presentation of the project is separate for each student. The discussion of the project is individual.</p> <p>The project can only be discussed after passing the written test.</p> <p>The project outline is assigned during the laboratory lessons (and simultaneously published on ADA).</p> <p>The project is considered passed if the student achieves a grade greater than 18/33 at the time of its discussion.</p> <p>The assigned project is valid only for the exam sessions held in the 2024-25 academic year.</p> <p>Final grade is the average of the grade obtained in the written test and project rounded up. A mark greater than 30 will result in the awarding of honours</p>
<b>Evaluation Criteria</b>	<ul style="list-style-type: none"> <li>● Knowledge and understanding: <ul style="list-style-type: none"> <li>○ Ability to understand the questions formulated for the written test and respond in a relevant and comprehensive manner</li> <li>○ Ability to understand the guidelines for carrying out laboratory and project activities</li> </ul> </li> <li>● Applied knowledge and understanding: <ul style="list-style-type: none"> <li>○ Exhaustive knowledge of the topics covered by the course and their use in carrying out exercises covered by the written test, discussion of theoretical issues and implementation of the project</li> </ul> </li> <li>● Independent judgment: <ul style="list-style-type: none"> <li>○ The written test includes three theory questions and/or exercises on topics included in the course program. Each question is scored a maximum of 11 points (based on the completeness and correctness of the answer). The test is considered passed if the student achieves at least 18/33 in the entire test.</li> <li>○ The project is developed following the guidelines provided during the laboratory lessons. The project must include a short description that outlines the implemented features and points of difference compared to the original project, UML class/package diagram (max 2 points), JAVADOC (max 2 points), system user guide and test cases (max 2 points), jar + script for jar execution (max 2 points), source code (15 points), extensions (10 points). The project is considered sufficient if rated at least 18/33</li> </ul> </li> <li>● Communication skills:</li> </ul>



	<p>o Ability to answer the questions formulated for the written test correctly, comprehensively and using technical language appropriately</p> <ul style="list-style-type: none"><li>● Ability to learn: Understanding of the course contents and ability to use the concepts learned in carrying out exercises and developing the software project in Java</li></ul>
Evaluation Measurement Criteria and Final Grade	<p>The final mark is the arithmetic mean of the marks obtained in the written test and project rounded up.</p>
<b>Additional information</b>	<p>Students are advised to rely exclusively on the information/communications provided on the official websites of the Department of Computer Science, or on social groups only if established and administered exclusively by the teachers of the relevant courses:</p> <ul style="list-style-type: none"><li>● <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea</a></li><li>● <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica">https://www.uniba.it/it/ricerca/dipartimenti/informatica</a></li><li>● <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a></li></ul> <p>The teaching programs are available here:</p> <ul style="list-style-type: none"><li>● <a href="https://programmi.di.uniba.it/">https://programmi.di.uniba.it/</a></li></ul> <p>The information that all students should know is written in the Teaching Regulations and study posters available on the site:</p> <ul style="list-style-type: none"><li>● <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea</a></li></ul> <p>Students are advised 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, ask the teacher for a meeting according to the reception procedures.</p> <hr/> <p>Students will be able to join the A.A. course forum. 2024/25 by enrolling in the course on the e-learning platform of the ADA department: <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a></p>