Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	**Programmazione II**
Corso di studio	Informatica e Tecnologie per la Produzione del Software
Anno Accademico	2023/24
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	09 CFU
Settore Scientifico Disciplinare	ING-INF/05
Lingua di erogazione	Italian
Anno di corso	Second
Periodo di erogazione	1st semester, the exact dates are specified in the schedule/regulations.
Obbligo di frequenza	Attendance is highly recommended.
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-tps-270/laurea-triennale-in-informatica-e-tecnologie-per-la-produzione-del-software-d.m.-270

Docente

Nome e cognome	Pasquale Ardimento
Indirizzo mail	pasquale.ardimento@uniba.it
Telefono	Not available
Sede	Department of Informatics, Via Orabona 4, 70125, Bari. Stanza n.569, V piano.
Sede virtuale	ADA Platform - https://elearning.uniba.it/course/index.php?categoryid=104
Sito web del docente	https://www.uniba.it/it/docenti/ardimento-pasquale
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Tuesday 3:00 PM - 5:00 PM, room 569 (Department of Computer Science) or by appointment.

Syllabus

Obiettivi formativi	The course aims to introduce the principles of the Object-Oriented paradigm, enabling students to acquire the skills to develop effective and efficient ways to solve

	problems using computers. It focuses on mastering the best methods for storing, accessing, representing, processing, and interpreting information through programming in the Java language.
Prerequisiti	<u>From the Programming course</u>: fundamentals of imperative programming, debugging skills, compilers; From the Computer Architectures and Operating Systems course: call stack, memory management, processes, and threads.
Contenuti di insegnamento (Programma)	Programming Environments and Languages (56 hours) • Introduction to Java applications, I/O, and operators (chapters 1-2) • Introduction and in-depth study of classes, objects, methods, and strings (chapters 3-6-8) • Flow control statements and logical operators (chapters 4-5) • Arrays and ArrayList (chapter 7) • Object-oriented programming: inheritance (chapter 9) • Object-oriented programming: polymorphism and interfaces (chapter 10) • Exception handling (chapter 11) • Strings, characters, and regular expressions (chapter 14) • File, I/O streams, NIO, and XML serialization (chapter 15) • Generic collections (chapter 16) • Lambdas and Streams (chapter 17) • Generic classes and methods: in-depth study (chapter 20) Guided exercises and laboratory (30 hours) • The Eclipse environment. • A single exercise that will be developed incrementally during the laboratory and will include: • Design and implementation of individual classes; • Design and implementation with multiple hierarchically organized classes and in packages; • Design and implementation of abstract classes and use of polymorphism; • Design and implementation with containers and exception handling; • • Design and implementation with file I/O.
Testi di riferimento	• Programmare in Java, undicesima edizione, di Paul J. Deitel, Harvey M. Deitel, editore Pearson, ISBN: 9788891916211 (capitoli 1-17, capitolo 20). Students who wish to do so can borrow the text from the Library. It may be convenient to check its availability through the University Library System at https://opac.uniba.it/easyweb/w8018/index.php? and contact the library to arrange the loan.
Note ai testi di riferimento	During the lessons, the instructor will use slides that cover the contents of the book; therefore, they will not be provided separately. The reference text includes all the topics of the course; therefore, it is recommended to study from the text and independently and consistently complete all the exercises at the end of each chapter covered in class. On the ADA platform of the department (see 'virtual venue' above), some written exam questions are available, along with examples of solved questions.

Organizzazione della didattica

Ore

Totali	Didattica frontale	Esercitazioni e Laboratorio	Studio individuale

86 ore	56 ore	30 ore		139 ore
CFU/ETCS				
09 CFU	07 CFU	02 CFU		

Metodi didattici	
	Classroom teaching with theory lessons, guided exercises in which students will interact with the instructor to solve exercises, and laboratory activities for programming. The laboratory activity aims to combine practice and theory for the development of software and problem-solving using the computer.

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	The student will acquire a basic understanding of Object-Oriented modeling and programming.
Conoscenza e capacità di comprensione applicate	Through the introduction of the Java programming language and the guided development of a project, the student will deepen their understanding of Object-Oriented programming, class composition, the use of class hierarchies, and some fundamental data structures. Additionally, the course covers the development of client-server applications and functional programming in Java.
Competenze trasversali	**Judgment Autonomy** The student will gain judgment autonomy concerning the implementation of Java software based on the principles of the Object-Oriented paradigm. **Ability to Learn Independently** The student will develop the ability to learn and navigate efficiently through issues that arise during the development of software implemented in Java in accordance with the principles of the Object-Oriented paradigm. **Communication Skills** The student will communicate appropriately in reference to the principles of Object-Oriented modeling and programming in Java.

Valutazione	
Modalità di verifica dell'apprendimento	**Prova d'esame**: To be taken in the laboratory. The exam consists of a questionnaire and two Java exercises. The time available for the exam is 90 minutes.

	Each Java exercise requires the implementation of a small Java application in the Eclipse development environment. Each Java exercise, if completed correctly and entirely, will be rated 7 (seven) points.

The questionnaire consists of 13 questions, with 12 multiple-choice questions and 1 open-ended question. Each multiple-choice question has four options, of which only one is correct. A correct answer is worth 1 point, and an incorrect answer is worth 0 points. The open-ended question is worth a maximum of six points.

All questions focus on theory and exercises related to topics in the syllabus.

The exam is considered passed if the student achieves a grade greater than or equal to 18/32.

Exemption tests
There are two exemption tests, both to be taken in the laboratory.

The first exemption test is held near the week of the break in lessons, typically around mid-November.
The second exemption test takes place immediately after the Christmas holidays before the first exam session of the academic year 2023-2024.

Both tests consist of a questionnaire and a Java exercise, with a time limit of 60 minutes for each test.

The Java exercise requires the implementation of a small Java application in the Eclipse development environment. The Java exercise, if completed correctly and entirely, will be rated 6 (six) points.

The questionnaire consists of 10 multiple-choice questions. Each question has four options, of which only one is correct. A correct answer is worth 1 point, and an incorrect answer is worth 0 points.

All questions and Java exercises focus on theory and exercises related to topics covered in the syllabus up to the moment of the test.

The first test is considered passed if the student achieves a grade greater than or equal to 9/16.

Passing the first test is necessary and sufficient to be admitted to the second test.
Both tests are considered passed if the arithmetic mean of the two tests is greater than or equal to 18/32.
Passing both exemption tests will count as passing the exam.

Final Score
The final grade is the sum of the scores obtained in the exam (or exemption test). Students who have achieved a score of 31/32 or 32/32 will be awarded honors (cum laude), equivalent to a grade of 30 with honors. |
| Criteri di valutazione | Knowledge and Understanding:
• Ability to understand the questions posed in the written exam and respond relevantly and comprehensively.
• Ability to understand the guidelines for carrying out laboratory activities and the project.
Applied Knowledge and Understanding:
• Comprehensive knowledge of the topics covered in the course and their application in solving exercises included in the written exam, addressing theoretical issues, and implementing the software project.
Judgment Autonomy: |

	• Student's ability to correct and validate the proper functioning of developed programs. Communicative Skills: • Ability to respond to questions posed in the written exam correctly, comprehensively, and using technical language appropriately. Learning Ability: • Understanding the course content and the ability to apply the learned concepts in solving exercises and developing the Java software project.
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	The final grade is the arithmetic sum of the scores obtained in the written exam and the project, rounded up.
Altro	Students are advised to rely exclusively on information/communications provided on the official websites of the Department of Computer Science or on social groups only if established and administered solely by the instructors of the respective courses: • 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.uniba.it/course/index.php?categoryid=104 The course programs are available here: • https://programmi.di.uniba.it/ The information that all students should be aware of is outlined in the educational regulations and study plans available on the website: • https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea/ Students are advised to be cautious of information circulating on unofficial websites or social groups, as they have often proven to be unreliable, incorrect, or incomplete.