UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

CONSIGLIO INTERCLASSE IN MATEMATICA

| General information | |
|---|---|
| Academic subject | ***Informatica*** |
| Degree course | *Matematica* |
| Academic Year | *2022/2023* |
| European Credit Transfer and Accumulation System (ECTS) | *6* |
| Language | *Italiano* |
| Academic calendar (starting and ending date) | *September 26th, 2021 – December 22th, 2022* |
| Attendance | *Not mandatory* |

| Professor/ Lecturer | |
|---|---|
| Name and Surname | *Marco Polignano* |
| E-mail | *marco.polignano@uniba.it* |
| Telephone | *+39 080 544 2312* |
| Department and address | *Office 759 Department of Computer Science, Campus Universitario,*<br>*Via E. Orabona 4, 70126 Bari* |
| Virtual headquarters | *Microsoft Teams (Marco Polignano – marco.polignano@uniba.it)* |
| Tutoring (time and day) | *Monday 16.30 -18.30*<br>*On Microsoft Teams*<br>*(for an optimal organization, please make an appointment by mail)* |

| ***Syllabus*** | |
|---|---|
| **Learning Objectives** | *Acquire basic concepts regarding computer architectures. Acquire basic concepts about programming methods and techniques. Apply these concepts to solve problems using the Python language. Basic notions and concepts about algorithms and computational complexity.* |
| **Course prerequisites** | *Elementary math concepts provided by upper secondary schools.* |
| **Contents** | *Introduzione al corso*<br>*Introduzione all'architettura dei calcolatori ed alle reti*<br>*Introduzione alla programmazione*<br>*Diagrammi di flusso*<br>*Linguaggi di programmazione*<br>*Programmazione strutturata*<br>*Variabili e funzioni*<br>*Strutture dati*<br>*Costrutti di programmazione*<br>*Introduzione alla computabilità*<br>*Complessità computazionale*<br>*Macchine di Turing*<br>*Introduzione al linguaggio*<br>*Python* |
| **Books and bibliography** | *Andrew J. Tanenbaum, Architettura dei calcolatori. Un approccio strutturale.*<br>*Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Introduzione agli algoritmi e strutture dati, McGraw-Hill Education*<br><br>*Slide ed altro materiale mostrato a lezione.* |
| **Additional materials** | *Books are recommended by the teacher but they are not mandatory. The teacher will provide students with supplemental material, such as scientific articles.* |

| **Work schedule** | | | |
|---|---|---|---|
| Total | Lectures | Hands on (Laboratory, working groups, seminars, field trips) | Out-of-class study hours/ Self-study hours |
| **Hours** | | | |
| *150* | *32* | *28* | *90* |
| **ECTS** | | | |
| *6* | *4* | *2* | |

| **Teaching strategy** | |
|---|---|
| | *Frontal classroom teaching and guided exercises. The e-learning mode will be applied only when requested by the Academic Senate as a measure to be taken to cope with the COVID-19 emergency.* |
| **Expected learning outcomes** | *At the end of the course, the student will be able to:* |
| **Knowledge and understanding on:** | ○ *Recognize components of the architecture of an electronic computer.*<br>○ *Formally describe a programming language.*<br>○ *Be familiar with the basics of programming.*<br>○ *Be familiar with the basics of computer science.* |
| **Applying knowledge and understanding on:** | ○ *Recognize and use data structures and primitive data types.*<br>○ *Use basic programming constructs.*<br>○ *Correctly use environments and tools to solve complex problems using a software program.* |
| **Soft skills** | • *Making informed judgments and choices*<br>○ *Decide which data structures and programming constructs should be used to solve complex problems.*<br>○ *Identify errors in software program code and correct them on their own.*<br>○ *Choose appropriate software and programming languages to solve complex problems.*<br>• *Communicating knowledge and understanding*<br>○ *Use Computer Science terminology correctly.*<br>• *Capacities to continue learning*<br>○ *Independently solve complex problems that require the use of a software program.*<br>○ *o Understand and resolve errors in software code.* |

| **Assessment and feedback** | |
|---|---|
| Methods of assessment | *Exam Sessions (after completion).*<br>*Examination with written exercises and theoretical open questions. The test will last 1.30h.* |
| Evaluation criteria | *The test will allow the student to be evaluated with respect to the following skills learned in the course.*<br>***Knowledge and understanding:***<br>*Ability to learn fundamental concepts regarding programming methods and techniques and general concepts of Computer Science.* |

| | |
|---|---|
| | ***Applying knowledge and understanding:***<br>*Ability to apply concepts learned to solve and implement solutions to a variety of problems.*<br>***Autonomy of Judgment:***<br>*Ability to judge the consistency of the logical structure used to construct algorithms. Ability to identify appropriate tools for solving programming problems.*<br>***Communication knowledge and understanding:***<br>*Learning the methods and languages of computer science to understand text, communicate learned knowledge, describe, analyze, and solve algorithmic problems.*<br>***Capacities to continue learning:***<br>*Acquire enduring learning methods supported by contextual research and by solving exercises and questions asked periodically throughout the course.* |
| Criteria for assessment and attribution of the final mark | *The final grade is given in thirtieths. The exam is considered passed when the grade is greater than or equal to 18. A score will be given for each individual question up to a total of 32 points. Honors will be awarded with scores above 30.* |
| **Additional information** | Attendance is recommended, in particular participation in the practical exercises, which also include the performance of classroom exercises. For non-attending students, it is suggested to consult the teacher during office hours to verify the correctness of the proposed solutions. |
| | |