COURSE DETAILS

"ADVANCED COMPUTER PROGRAMMING"

SSD ING-INF/05

DEGREE PROGRAMME: BACHELOR DEGREE IN COMPUTER ENGINEERING

ACADEMIC YEAR: 2023-2024

GENERAL INFORMATION – TEACHER REFERENCES

TEACHER: RAFFAELE DELLA CORTE
PHONE: 0817683820
EMAIL: raffaele.dellacorte2@unina.it

GENERAL INFORMATION ABOUT THE COURSE

INTEGRATED COURSE (IF APPLICABLE): N.A.
MODULE (IF APPLICABLE): N.A.
CHANNEL (IF APPLICABLE): N.A.
YEAR OF THE DEGREE PROGRAMME (I, II, III): III
SEMESTER (I, II): II
CFU: 9

REQUIRED PRELIMINARY COURSES (IF MENTIONED IN THE COURSE STRUCTURE “REGOLAMENTO”)

Programmazione.

PREREQUISITES (IF APPLICABLE)

None.

LEARNING GOALS

The course aims at providing students with advanced knowledge and expertise related to concurrent and distributed programming, introducing the tools to develop and debug multithreading and network applications using Java and Python, and introducing the concept of middleware and of the different solutions used in industry, focusing on both the message- oriented and service-oriented models, with application on real technology.
The course also introduces the concepts and tools for web-application development, from both the fornt-end and back-end perspective.

EXPECTED LEARNING OUTCOMES (DUBLIN DESCRIPTORS)

Knowledge and understanding

The student needs to demonstrate to know and understand the problems concerning concurrent and distributed programming, mainly in Java and Python, as well as the characteristics of the different middleware technologies. The course provides students with knowledge and methodological, theoretical and practical tools needed to recognize, analyze and resolve problems related to the development of multithreading and network applications, allowing the studentsto master the development of advanced software projects in Java and Python.

Applying knowledge and understanding

The student needs to show ability to solve problems related to the concurrent and distributed programming, using the methodological, theoretical and practical competences concerning the advanced computer programming introduced during the course, in order to develop multithreading and network software projects in Java and Python.

COURSE CONTENT/SYLLABUS

Concurrent and network programming in Java.

Java overview and collection classes. Concurrent programming in java. Threads in java, states of a threads, threads pool.Synchronization in Java. Java monitor and the package java.util.concurrent of Java 1.5. Build automation tool and Java debugging. Generic programming, reflection and annotation Java.
Network programming on Java network. The java.net package. Socket TCP in Java: Socket and ServerSocket classes. Socket UDP in Java: DatagramSocket and DatagramPacket classes. Server multithread. Remote object abstraction. Proxy-Skeleton.

Middleware models.

Definition and properties of the middleware level. Enterprise Application Integration (EAI). Remote Procedure Call (RPC),Message Oriented (MOM), Transaction Processing (TP), Tuple Space (TS), Remote Data Access (RDA), Distributed Object(DOM), Component model (CM), web services, microservices.

Object oriented model.

Java Message Service (JMS) specification, client and provider. Point-to-point and publish-subscribe communication. Modellodi programmazione JMS. Messaggi JMS ed aspetti avanzati.

Service model and web-app application development.

Introduction to SOAP and RPC service. RESTful Web Services, resource and Uniform Resource Identifier (URI). RESTful services and HTTP methods. RESTful Web Services development with Java framework and web-app development with

HTML, Javascript and front-end development framework.

The Python language.
Data type, control statements, argument passing, data collection, file, functions, modules and debugging in Python. Object-oriented programming in Python: classes, objects, inheritance, polymorphism, abstract classes. Concurrent and network programming in Python: multithreading, synchronization, socket. Multi-language integration example. Data Science in Python.

READINGS/BIBLIOGRAPHY

- Slides of the course.
- Books:
  - B. Eckel, *“Thinking in Java”.*
  - L. H. Etzkorn, *"Introduction to Middleware - Web Services, Object Components, and Cloud Computing".*
  - Semmy Purewal, *“Learning Web App Development”.*
  - Craig Walls, *“Spring in Action”.*
  - Allen B. Downey, *“Think Python”.*
  - Mark Lutz, *“Programming Python”.*
- Training material.
- On-line resources.

TEACHING METHODS

Lectures and practical exercises.

EXAMINATION/EVALUATION CRITERIA

a) Exam type:

Exam type	
written and oral	X
only written	
only oral	
project discussion	
other	Computer-based coding exam

In case of a written exam, questions refer to:	**Multiple choice answers**	
	Open answers	
	Numerical exercises	

b) Evaluation pattern:

The written exam performance is binding to have access to the oral exam.Midterm evaluations could be scheduled in place of the written exam.