
Ano Letivo 2023-24

Unidade Curricular PROGRAMAÇÃO ORIENTADA A OBJETOS

Cursos ENGENHARIA DE SISTEMAS E TECNOLOGIAS INFORMÁTICAS (1.º ciclo)

Unidade Orgânica Instituto Superior de Engenharia

Código da Unidade Curricular 19411006

Área Científica CIÊNCIAS INFORMÁTICAS

Sigla

Código CNAEF (3 dígitos) 481

Contributo para os Objetivos de
Desenvolvimento Sustentável - 4, 8, 9
ODS (Indicar até 3 objetivos)

Línguas de Aprendizagem Português

Modalidade de ensino

Presencial

Docente Responsável

Jaime Afonso Do Nascimento Carvalho Martins

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
JAIME AFONSO DO NASCIMENTO CARVALHO MARTINS	PL; TP	TP1; PL1	14TP; 42PL

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
1º	S2	14TP; 42PL	130	5

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Frequência da UC de Programação

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Proporcionar uma visão geral sobre os conceitos definidos em programação orientada a objetos (POO). Capacitar o aluno a oferecer soluções algorítmicas para problemas computacionais. Utilizar uma linguagem de POO para implementação das soluções.

Conteúdos programáticos

Introdução ao paradigma da Programação Orientada a Objetos

1. Perspetiva histórica
2. Características da programação orientada a objetos (encapsulamento, partilha de comportamento, evolução)
3. Objetos (noção de objeto e identidade; protocolo de utilização de um objeto; mensagens e métodos; noção de estado e comportamento).
4. Linguagens baseadas em classes; instanciação de objetos.
5. Hierarquias de objetos: Mecanismos de herança; Reutilização de código; Sistemas reflexivos; Polimorfismo.
6. Interfaces: Separação entre interface e implementação de interfaces.
7. Relações entre objetos: Extensão, generalização e especialização; Associação, agregação e composição; Coesão e acoplamento.
8. Aplicações.

Metodologias de ensino (avaliação incluída)

Metodologias de ensino

Aulas Teórico-Práticas: Método semi-expositivo, orientado de acordo com um ensino baseado em projetos (*project based learning*), com introdução e investigação dos objetivos e conteúdos correspondentes a cada projeto.

Aulas Práticas: Prática laboratorial em computador. Análise, resolução e codificação de projetos, selecionados em conformidade com o conteúdo teórico semanal.

Avaliação

Componentes da avaliação classificadas de 0-20 valores:

- Trabalho(s) prático(s) (TP)
- Prova escrita (PE): Teste / Exame

Classificação mínima em cada uma das componentes 8 valores.

Nota final: 0.5 PE + 0.5 TP

Bibliografia principal

1. Hunt, J. (2021). A beginners guide to Python 3 programming. Springer
2. Lott, S., Phillips, D. (2021). Python Object-Oriented Programming: Build robust and maintainable object-oriented Python applications and libraries. Packt.
3. Lutz, M. (2013). Learning Python: Powerful object-oriented programming . " O'Reilly Media, Inc.".
4. Reges, S., Stepp, M., & Obourn, A. (2018). Building Python Programs . Pearson.
5. Romano, F. (2015). Learning Python. Packt Publishing.
6. Sage, K. (2019). Concise Guide to Object-Oriented Programming: An Accessible Approach Using Java. Springer
7. Summerfield, M. (2008), Programming in Python 3: A Complete Introduction to the Python Language. Addison-Wesley Professional.

Academic Year 2023-24

Course unit OBJECT ORIENTED PROGRAMMING

Courses SYSTEMS ENGINEERING AND COMPUTER TECHNOLOGIES (1st cycle)

Faculty / School INSTITUTE OF ENGINEERING

Main Scientific Area

Acronym

CNAEF code (3 digits) 481

Contribution to Sustainable Development Goals - SGD (Designate up to 3 objectives) 4, 8, 9

Language of instruction Portuguese

Teaching/Learning modality Classroom-based learning

Coordinating teacher JAIME AFONSO DO NASCIMENTO CARVALHO MARTINS

Teaching staff	Type	Classes	Hours (*)
JAIME AFONSO DO NASCIMENTO CARVALHO MARTINS	PL; TP	TP1; PL1	14TP; 42PL

* For classes taught jointly, it is only accounted the workload of one.

Contact hours	T	TP	PL	TC	S	E	OT	O	Total
	0	14	42	0	0	0	0	0	130

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

Frequency of the Programming CU

The students intended learning outcomes (knowledge, skills and competences)

Provide an overview of the concepts defined in object-oriented programming (POO). Empower the student to offer algorithmic solutions to computational problems. Use a POO language to implement solutions

Syllabus

Introduction to the Paradigm of Object-Oriented Programming

1. Historical perspective
2. Characteristics of object-oriented programming (encapsulation, behavior sharing, evolution)
3. Objects (notion of Object and identity; protocol of use of an object; messages and methods; notion of state and behavior.
4. Class-based languages; instantiation of Objects.
5. Object Hierarchies: Inheritance Mechanisms; Code reuse; Reflective systems; Polymorphism.
6. Interfaces: Separation between interface and implementation of interfaces.
7. Relationships between Objects: Extension, generalization and specialization; Association, aggregation and composition; Cohesion and coupling.
8. Applications.

Teaching methodologies (including evaluation)

Teaching methodologies

- Theoretical-Practical Classes: Introduction to necessary concepts, oriented according to project-based learning, with introduction and investigation of the objectives and contents corresponding to each project.
- Practical Classes: Laboratory practice in computers. Analysis, resolution, and coding of projects, selected in accordance with weekly theoretical content.

Evaluation

Evaluation components classified from 0-20 points:

1. Practical work(s) (TP)
1. Written test (PE): Test / Exam

Minimum classification in each of the components: 8 points.

Final grade: 0.5 PE + 0.5 TP

Main Bibliography

1. Hunt, J. (2021). A beginners guide to Python 3 programming. Springer
2. Lott, S., Phillips, D. (2021). Python Object-Oriented Programming: Build robust and maintainable object-oriented Python applications and libraries. Packt.
3. Lutz, M. (2013). Learning Python: Powerful object-oriented programming . " O'Reilly Media, Inc.".
4. Reges, S., Stepp, M., & Obourn, A. (2018). Building Python Programs . Pearson.
5. Romano, F. (2015). Learning Python. Packt Publishing.
6. Sage, K. (2019). Concise Guide to Object-Oriented Programming: An Accessible Approach Using Java. Springer
7. Summerfield, M. (2008), Programming in Python 3: A Complete Introduction to the Python Language. Addison-Wesley Professional.