



1. Caracterização da Unidade Curricular

1.1 Designação

[3080] Programação Orientada por Objetos / Object Oriented Programming

1.2 Sigla da área científica em que se insere

CE

1.3 Duração

Unidade Curricular Semestral

1.4 Horas de trabalho

162h 00m

1.5 Horas de contacto

Total: 72h 30m das quais TP: 67h 30m | O: 5h 00m

1.6 ECTS

6

1.7 Observações

Unidade Curricular Obrigatória

2. Docente responsável

Não existe docente responsável para esta unidade curricular

3. Docentes e respetivas cargas letivas na unidade curricular Não existem docentes definidos para esta unidade curricular



**4. Objetivos de aprendizagem
(conhecimentos, aptidões e
competências a desenvolver
pelos estudantes)**

1. Utilização da linguagem de programação Python, na sua vertente de programação orientada por objetos.
2. Desenvolvimento de aplicações, recorrendo ao paradigma da programação orientada por objetos

e ao encapsulamento de dados.
3. Adquirir conhecimento sobre as construções da linguagem Python, na sua vertente de programação orientada por objetos.
4. Definir e utilizar classes e classes derivadas e outros conceitos relacionados com objetos.
5. Conceber e escrever programas que resolvam problemas recorrendo ao paradigma da programação orientada por objetos.
6. Produzir documentação técnica para os programas desenvolvidos.
7. Utilizar de forma adequada as ferramentas de desenvolvimento e de produção de documentação.

**4. Intended learning outcomes
(knowledge, skills and
competences to be developed
by the students)**

1. Use the Python programming language, in its object-oriented programming paradigm.
2. Application development, using the object-oriented programming paradigm and data encapsulation.
3. Acquire knowledge about the constructions of the Python language, in its object-oriented programming paradigm.
4. Define and use classes and derived classes and other concepts related to objects.
5. Design and write programs that solve problems using the object-oriented programming paradigm.
6. Produce technical documentation for the developed programs.
7. Properly use development and documentation production tools.

5. Conteúdos programáticos

1. Classes, objetos, atributos de dados ? membros, atributos funções ? métodos, encapsulamento.
2. Construtor, ?self?.
3. Redefinição de operadores.
4. ?instance?, herança, classe base, classe derivada, super, polimorfismo.
5. Exceções.

5. Syllabus

1. Classes, objects, data attributes - members, function attributes - methods, encapsulation.
2. Constructor, ?self?.
3. Operator overloading.
4. ?instance?, inheritance, base class, derived class, super, polymorphism.
5. Exceptions.

6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular

Esta unidade curricular aprofunda a programação de computadores fazendo a transição da programação procedimental para a programação orientada por objetos, através da linguagem de programação Python, na sua vertente de programação orientada por objetos. Os conteúdos programáticos constituem uma introdução à programação orientada por objetos em Python. Os objetivos de aprendizagem estão fortemente relacionados entre si e são abrangidos pela totalidade dos conteúdos programáticos.

6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes

This course deepens computer programming by making the transition from procedural programming to object-oriented programming, through the Python programming language, in its object-oriented programming paradigm. The syllabus is an introduction to object-oriented programming in Python. The learning outcomes are strongly related to each other and are covered by the entire syllabus.

**7. Metodologias de ensino
(avaliação incluída)**

Ensino teórico-prático. As aulas compreendem exposição dos conceitos teóricos, resolução de exercícios e programação de computador. Existem entre 20 e 30 trabalhos, sem relatório, realizados individualmente ao longo do semestre.

A unidade curricular tem avaliação distribuída com exame final. Os conceitos teóricos, a resolução de exercícios e a programação de computador são avaliados em exame (NE) e nos trabalhos realizados individualmente ao longo do semestre (NT). A nota (NT) é a média aritmética dos trabalhos. O exame é constituído por uma prova escrita e por uma prova oral. Os alunos com classificação superior a 8 valores na prova escrita, têm acesso à prova oral. A nota do exame (NE) é a nota da prova oral ou, caso o estudante seja dispensado desta e não a queira realizar, é a nota da prova escrita. A nota final (NF) é obtida por $NF=0,5*NE+0,5*NT$. Para obter aprovação é preciso ter nota mínima de 9,5 valores em NE e NF.

**7. Teaching methodologies
(including assessment)**

Theoretical-practical teaching. Classes include exposure of theoretical concepts, exercise resolution and computer programming. There are between 20 and 30 individual assignments, with no report, throughout the semester.

The curricular unit has distributed assessment with a final exam. Theoretical concepts, exercise resolution and computer programming are evaluated in exam (NE) and in the assignments carried out individually throughout the semester (NT). The grade (NT) is the average of all works. The exam has a written part and an oral part. Students with a classification higher than 8 in the written part have access to the oral part. The exam grade (NE) is the oral test grade or, if the student is exempt from this and does not want to take it, it is the written test grade. The final grade (NF) is obtained by $NF=0,5*NE+0,5*NT$. To obtain approval it is necessary to have a minimum grade of 9,5 values in NE and NF.

**8. Demonstração da coerência
das metodologias de ensino
com os objetivos de
aprendizagem da unidade
curricular**

Os objetivo de aprendizagem 1 a 7 são suportado pelas aulas teóricas, pelas aulas de resolução de exercícios , pelas aulas de programação e pelos trabalhos realizados ao longo do semestre.

**8. Evidence of the teaching
methodologies coherence with
the curricular unit's intended
learning outcomes**

Learning outcomes 1 to 7 are supported by theoretical classes, exercise solving classes, programming classes and assignments carried out throughout the semester.

**9. Bibliografia de
consulta/existência obrigatória**

Swaroop C. H., A Byte of Python. (Livro livre disponível em <https://python.swaroopch.com/>)

Allen B. Downey, Think Python, 2nd Edition, Green Tea Press, 2015. (Livro livre disponível em <https://greenteapress.com/wp/think-python-2e/>)

Joel Grus, Data Science from Scratch: First Principles with Python, 2nd Edition. O'Reilly, 2019.

10. Data de aprovação em CTC 2024-07-17

11. Data de aprovação em CP 2024-06-26