

TeSP - Animação e Modelação 3D

Técnico Superior Profissional

Plano: Aviso n.º 11775/2016 - 27/09/2016

Ficha da Unidade Curricular: Linguagens de Programação

ECTS: 6; Horas - Totais: 162.0, Contacto e Tipologia, TP:15.0; PL:45.0;

Ano | Semestre: 1 | S2

Tipo: Obrigatória; Interação: Presencial; Código: 624013

Área de educação e formação: Ciências informáticas

Docente Responsável

Ricardo Nuno Taborda Campos

Professor Adjunto

Docente(s)

Ricardo Nuno Taborda Campos

Professor Adjunto

Objetivos de Aprendizagem

Esta unidade curricular tem por objectivo introduzir os alunos à algoritmia e à programação de computadores com recurso a linguagens de programação de alto nível, nomeadamente Python.

Objetivos de Aprendizagem (detalhado)

Esta unidade curricular tem por objectivo introduzir os alunos à programação de computadores de alto nível, nomeadamente através da linguagem de programação Python. Ao concluir esta unidade o estudante deverá:

1. Conhecer e compreender os fundamentos básicos da computação e dos computadores;
2. Ser capaz de projetar algoritmos para a resolução de problemas usando linguagem natural, fluxogramas e pseudo-linguagem;
3. Conhecer e compreender os princípios da programação estruturada;
4. Saber estruturar e escrever programas com recurso a linguagens de programação de alto nível, nomeadamente Python. No final da UC o aluno deverá (a) ter conhecimento profundo das características desta linguagem de programação e saber instalar e configurar o ambiente de

desenvolvimento; (b) ter conhecimento dos principais comandos da linguagem de programação; (c) ter conhecimento das principais bibliotecas existentes nesta linguagem de programação; (d) saber automatizar rotinas com recurso a estruturas de controlo e iteração; (e) saber escrever e estruturar programas com recurso a arrays; estruturas de dados avançadas e compreensão de listas; (f) ser capaz de escrever e guardar ficheiros de texto, CSV, JSON e documentos web; (g) saber decompor problemas em sub-tarefas com recurso a funções reutilizáveis e anónimas; (h) saber criar e disponibilizar módulos;

Conteúdos Programáticos

1. Introdução à Ciência dos Computadores
2. Algoritmos
3. Introdução à Programação Estruturada
4. Linguagens de Programação de Alto Nível: Python

Conteúdos Programáticos (detalhado)

1. Introdução à Ciência dos Computadores
 - 1.1. O que é um Computador?
 - 1.2. O que é a Ciência dos Computadores?
 - 1.3. Breve História da Computadores.
 - 1.3.1. Linha do tempo.
 - 1.3.2. Alan Turing, Máquina de Turing, Teste de Turing.
 - 1.3.3. Lei de Moore.
 - 1.3.4. Arquitetura de von Newman.
 - 1.3.5. Memória do Computador
 - 1.4. Representação da Informação
 - 1.5. Tendências Atuais (Inteligência Artificial; Machine Learning; Big Data; Ciência dos Dados; Internet das Coisas; Computação em Nuvem; Processamento da Linguagem Natural).
 - 1.6. Porquê aprender a programar?
 - 1.7. O que é um programa?
 - 1.8. Linguagens de programação de baixo nível e de alto nível
 - 1.9. Compiladores vs Interpretadores
2. Algoritmos
 - 2.1. O que é um algoritmo?
 - 2.2. Características dos algoritmos.
 - 2.3. Como resolver problemas através de algoritmos?
 - 2.4. Como desenhar algoritmos? Linguagem natural, fluxogramas e pseudo-linguagem.
3. Introdução à Programação Estruturada
 - 3.1. Estruturas de Sequência.
 - 3.2. Estruturas de Seleção.
 - 3.3. Estruturas de Repetição.
4. Linguagens de Programação de Alto Nível: Python
 - 4.1 Programação em Python

- 4.1.1. Porquê programar em Python?
- 4.1.2. História do Python
- 4.1.3. Características
- 4.1.4. Vantagens
- 4.1.5. Instalação do Ambiente de Desenvolvimento

- 4.2. Introdução ao Python
 - 4.2.1. Comentários
 - 4.2.2. Ajuda no Python
 - 4.2.3. Entrada e saída de informação
 - 4.2.4. Variáveis
 - 4.2.5. Tipos de dados
 - 4.2.6. Operadores
 - 4.2.7. Operações de Casting
 - 4.2.8. Formatação de dados
 - 4.2.9. Imutabilidade vs Mutabilidade
 - 4.2.10. Notebooks em Python

- 4.3. Importação e Utilização de Bibliotecas
 - 4.3.1. Módulos internos
 - 4.3.2. Módulos externos
 - 4.3.3. Módulos frequentemente usados

- 4.4. Estruturas de Seleção e Repetição
 - 4.4.1. IF
 - 4.4.2. For
 - 4.4.3. While
 - 4.4.4. Break/Continue

- 4.5. Estruturas de Dados Simples
 - 4.5.1. Arrays
 - 4.5.2. Arrays Multidimensionais / Matrizes
 - 4.5.3. Jagged Arrays

- 4.6. Estruturas de Dados Avançadas
 - 4.6.1. Listas
 - 4.6.2. Conjuntos
 - 4.6.3. Dicionários
 - 4.6.4. Tuples
 - 4.6.5. Named Tuples
 - 4.6.6. Enums

- 4.7. Compreensão de Listas e LINQ
 - 4.7.1. Introdução à compreensão de listas
 - 4.7.2. LINQ em Python

- 4.8. Leitura e Escrita de Ficheiros
 - 4.8.1. Ficheiros de texto

4.8.2. Ficheiros de Imagem

4.8.3. Ficheiros PDF

4.8.4. Ficheiros MS Word

4.8.5. Ficheiros HTML

4.8.6. Ficheiros CSV

4.8.7. Ficheiros JSON

4.9. Funções

4.9.1 Funções definidas pelo utilizador

4.9.2. Funções geradoras

4.9.3. Funções lambda (MAP, Filter, Reduce)

4.9.4. Agendamento de Tarefas e de Scripts envolvendo Funções

4.10. Criação e Partilha de Pacotes de Software

4.10.1. Introdução ao Github

4.10.2. Introdução aos sistemas de controlo de versão (GIT)

4.10.3. Criação e distribuição local e online (PyPi e Github) de módulos

4.10.4. Criação e distribuição local e online (PyPi e Github) de pacotes de software

Metodologias de avaliação

Avaliação por Frequência: Frequência I (40%) [Prova com consulta] + Frequência II (60%) [Prova com consulta parcial dos conteúdos]. Participação obrigatória em todos os elementos de avaliação (nota mínima de 6 valores em cada um).

Avaliação por Exame: Exame (100%) [Prova sem consulta].

Requisitos de admissibilidade à frequência e ao exame:

- Mínimo de 70% de assiduidade às aulas (exceto trabalhadores estudantes);
- Mínimo de 80% na entrega dos problemas de programação propostos nas aulas;
- As presenças em aula ou a resolução dos problemas não são classificados com nota nem contam para avaliação, constituem, no entanto, condição necessária para aprovação à UC por frequência e exame. O incumprimento de qualquer um destes itens impede o aluno de se submeter à frequência e ao exame.

Software utilizado em aula

Python - Anaconda

Jupyter Notebooks

Git

PyCharm

Moodle: plataforma de eLearning do IPT

Estágio

Não aplicável.

Bibliografia recomendada

- Costa, E. e , . (2016). *Programação em Python - Fundamentos e Resolução de Problemas* . 1, FCA. Lisboa
- Liang, D. (2012). *Introducing to Programming using Python* . 1, NA. NA
- Downey, A. (0). *Think Python - How to Think Like a Computer Scientist* Acedido em 16 de fevereiro de 2018 em <http://greenteapress.com/wp/think-python>
- Severance, C. (0). *Python for Everybody - Exploring Data Using Python 3* Acedido em 16 de fevereiro de 2018 em http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf

Coerência dos conteúdos programáticos com os objetivos

Os conteúdos programáticos estão em coerência com os objetivos da unidade curricular, atendendo a que:

- O ponto 1 dos conteúdos programáticos pretende concretizar o ponto 1 dos objetivos
- O ponto 2 dos conteúdos programáticos pretende concretizar o ponto 2 dos objetivos
- O ponto 3 dos conteúdos programáticos pretende concretizar o ponto 3 dos objetivos
- O ponto 4 dos conteúdos programáticos pretende concretizar o ponto 4 dos objetivos

Metodologias de ensino

Aulas teórico-práticas expositivas onde se descrevem os conceitos fundamentais com recurso a meios áudio-visuais e a exemplos práticos.

Aulas práticas de resolução de casos práticos e aplicação dos conceitos a cenários de utilização real.

Coerência das metodologias de ensino com os objetivos

Os objetivos de aprendizagem do curso são atingidos através da realização de um conjunto de exercícios práticos permitindo desta forma que os alunos solidifiquem as competências adquiridas.

Língua de ensino

Português

Pré-requisitos

Não aplicável.

Programas Opcionais recomendados

Não aplicável.

Observações

Os conteúdos da UC serão trabalhados tendo em vista o cumprimento dos objetivos de desenvolvimento sustentável (ODS), em particular o item "Educação de Qualidade".

Docente responsável

Ricardo
Campos

Assinado de forma
digital por Ricardo
Campos
Dados: 2021.03.03
12:50:38 Z
