

Ano Lectivo: 2005/2006

Disciplina: Algoritmia e Programação

Docente: José Manuel Torres

Curso - Ano:

Engenharia Informática - 1º Ano

Regime:

Semestral – S1

Domínio: Específica

Horário Semanal:

Quatro Horas (2 aulas de 2 horas)

Enquadramento e Objectivos da Disciplina:

Estudo da noção de algoritmo, seu desenvolvimento e descrição utilizando notações estruturadas. Introdução à metodologia da utilização de computadores na resolução de problemas. Estudo de conceitos de programação como estruturas de controlo, tipos e declarações de dados, operadores e expressões. Introdução a estruturas de dados básicas como vectores, matrizes e cadeias de caracteres. Utilização de procedimentos e funções como forma de estruturação de um programa. Utilização de tipos estruturados e ficheiros. Estudo da linguagem procedimental C.

Sistema de Avaliação:

1- Avaliação Contínua:

Os conhecimentos do aluno serão certificados através da realização de dois testes escritos sem consulta, com igual peso na nota final, cujas datas serão marcadas pelo docente no início do semestre abrangendo no seu conjunto a totalidade do programa e tendo no total um peso de 70% na nota final.

Os alunos terão ainda de realizar um trabalho prático em grupo a ser entregue até ao último dia de aulas e que terá um peso de 30% na nota final.

Cada falha a um momento da avaliação será convertida numa classificação zero para efeitos de cálculo da média final. Uma classificação média final inferior a 9,5 valores remete o aluno automaticamente para o exame de recurso (não estando prevista a realização de provas orais).

A assistência às aulas teóricas, teórico-práticas ou práticas não laboratoriais é obrigatória em, pelo menos, 40% das aulas dadas. Com excepção dos trabalhadores-estudantes, o não cumprimento das percentagens indicadas impede o aluno de ser avaliado à disciplina.

2- Época de Recurso/Época especial:

O exame de recurso/época especial é uma prova que incide sobre a totalidade da matéria teórico-prática incluída no programa da disciplina. Esta prova destina-se aos alunos que não obtiveram aprovação na disciplina na época normal, ou alunos em regimes especiais.

Caso o aluno não tenha realizado trabalho prático à disciplina a nota final será dada pela nota do exame de recurso.

Caso o aluno tenha realizado trabalho prático à disciplina a nota final contará com a nota do trabalho de acordo com a fórmula usada na avaliação contínua (30% trabalho + 70% exame) desde que a nota do trabalho seja superior à nota do exame. Se a nota do trabalho for inferior à nota do exame, a nota final do aluno será a nota obtida no exame de recurso.

Programa da Disciplina:

1. Algoritmos e Computação
 - 1.1 Noção de Algoritmo
 - 1.2 Representação de Algoritmos
 - 1.2.1 Linguagem auxiliar (pseudo-código)
 - 1.2.2 Diagrama de fluxo (fluxograma)
 - 1.2.3 Linguagem de programação
 - 1.3 Prova e Teste de Algoritmos
 - 1.4 Programação "top-down" e "bottom-up"
 - 1.5 Resolução de Problemas
 - 1.5.1 Análise do Problema
 - 1.5.2 Especificação do Programa
 - 1.5.3 Codificação do Programa
 - 1.5.4 Teste e Refinamento do Programa
 - 1.5.5 Geração da Documentação
 - 1.5.6 Manutenção do Programa
2. Introdução à Programação
 - 2.1 Conceitos de programa e software
 - 2.2 Tipos de software
 - 2.3 Linguagem máquina, assembly e de alto nível
 - 2.4 Programação em linguagens de alto nível
 - 2.4.1 Programa fonte
 - 2.4.2 Compilador
 - 2.4.3 Interpretador
 - 2.4.4 Programa objecto
 - 2.4.5 Bibliotecas
 - 2.4.6 Programa executável
 - 2.5 Principais linguagens de programação de alto nível e suas aplicações
3. Tipos e declarações de dados, expressões e funções intrínsecas
 - 3.1 Tipos de dados
 - 3.1.1 Tipos ordinais
 - 3.1.2 Tipos reais
 - 3.1.3 Tipo cadeia de caracteres
 - 3.1.4 Definição e conversão de tipos
 - 3.2 Declaração de constantes
 - 3.3 Operadores e expressões
 - 3.3.1 Definição de operando, operador e expressão
 - 3.3.2 Operadores aritméticos
 - 3.3.3 Operadores lógicos
 - 3.3.4 Operadores booleanos
 - 3.3.5 Operadores relacionais
 - 3.3.6 Regras de precedência na avaliação de expressões
 - 3.4 Funções intrínsecas
4. Estruturas de controlo
 - 4.1 Instruções simples e estruturadas
 - 4.2 Instruções condicionais
 - 4.2.1 Instrução "if-then-else"

- 4.2.2 Instrução de selecção "case"
- 4.3 Instruções repetitivas
 - 4.3.1 Instrução "while"
 - 4.3.2 Instrução "for"
- 4.4 Instruções "goto" e etiquetas (*labels*)
- 4.5 Estruturas embutidas
- 5. Procedimentos e funções
 - 5.1 Conceitos de procedimento e função
 - 5.2 Variáveis locais vs globais
 - 5.3 Procedimentos
 - 5.3.1 Introdução aos procedimentos
 - 5.3.2 Tipos de parâmetros
 - 5.3.3 Passagem de parâmetros por valor
 - 5.3.4 Passagem de parâmetros por referência
 - 5.4 Funções
 - 5.4.1 Introdução às funções
 - 5.4.2 Diferenças relativamente aos procedimentos
 - 5.4.3 Parâmetros de entrada
 - 5.4.4 Parâmetro de retorno
 - 5.5 Introdução à recursividade
- 6. Variáveis indexadas e cadeias de caracteres
 - 6.1 Introdução à indexação
 - 6.2 Conceitos de vector e matriz
 - 6.3 Utilização de vectores
 - 6.3.1 Pesquisa em vectores: linear e binária
 - 6.3.2 Ordenação de vectores: por selecção, por fusão, "Bubble sort", por partição e troca
 - 6.4 Utilização de matrizes
 - 6.5 Utilização de cadeias de caracteres
 - 6.5.1 Conceito de cadeia de caracteres
 - 6.5.2 Operações sobre cadeias de caracteres
- 7. Tipos Estruturados
 - 7.1 Noção de tipos estruturados
 - 7.2 Registos definidos pelo utilizador
 - 7.2.1 Noção de registo
 - 7.2.2 Operações com registos
- 8. Ficheiros
 - 8.1 Noção de ficheiro
 - 8.2 Tipos de ficheiros
 - 8.3 Utilização de um ficheiro
 - 8.3.1 Declaração de um ficheiro
 - 8.3.2 Associação com uma variável
 - 8.3.3 Abertura do ficheiro
 - 8.3.4 Operações comuns com ficheiros
 - 8.3.5 Fecho de um ficheiro
 - 8.4 Utilização de ficheiros de texto
 - 8.5 Utilização de ficheiros de acesso directo

Bibliografia Principal:

- [1] Marques de Sá, J. P. "Fundamentos de Programação Usando C", Editora FCA, 2004, ISBN 972-722-475-X .
- [2] Damas, Luis "Linguagem C", Editora FCA, 1999, ISBN 972-722-156-4
- [3] Sampaio, Isabel; Sampaio, Alberto "Fundamental da Programação em C", Editora FCA, 1998, ISBN 972-722-130-0

Bibliografia Complementar:

- [4] Kernighan, Brian W. "The C Programming Language", Prentice-Hall International, 1988
- [5] Cormen, Leiserson, Rivest "Introduction to Algorithms " MIT-Press, 1990, ISBN: 0-262-03141-8 (biblioteca UFP).
- [6] Abelson, H., Sussman, G. "Structure and Interpretation of Computer Programs" MIT-Press, 2000, ISBN: 0-262-51087-1 (url: mitpress.mit.edu/sicp/full-text/book/book.html)
- [7] Pressman, Roger S. "Software Engineering – A Practitioner’s Approach – fourth edition" McGraw-Hill, 1997, ISBN: 0-07-052182-4 (biblioteca UFP).

Distribuição de Tempos Lectivos e da Bibliografia:

- 1. Algoritmos e Computação.
Horas Previstas: 8h
Bibliografia: [1,2,3]
- 2. Introdução à Programação.
Horas Previstas: 4h
Bibliografia: [1,2,3]
- 3. Tipos e declarações de dados, expressões e funções intrínsecas.
Horas Previstas: 4h
Bibliografia: [1,2,3]
- 4. Estruturas de controlo.
Horas Previstas: 8h
Bibliografia: [1,2,3]
- 5. Procedimentos e funções.
Horas Previstas: 8h
Bibliografia: [1,2,3]
- 6. Variáveis indexadas e cadeias de caracteres
Horas Previstas: 8h
Bibliografia: [1,2,3]
- 7. Tipos Estruturados.
Horas Previstas: 6h
Bibliografia: [1,2,3]
- 8. Ficheiros.
Horas Previstas: 6h
Bibliografia: [1,2,3]

As restantes horas serão dedicadas a revisões para avaliação.

Horário de Atendimento ao Aluno: A designar

Resumo:

Algoritmos e Computação. Introdução à Programação. Tipos e declarações de dados, expressões e funções intrínsecas. Estruturas de controlo. Procedimentos e funções. Variáveis indexadas e cadeias de caracteres. Tipos Estruturados. Ficheiros.

Abstract:

Algorithms and Computation. Introduction to Programming. Data types and their declaration, expressions and built-in functions. Control structures. Procedures and functions. Indexed variables and strings. Structured Types. Files.