

PLANO DE ENSINO - 2º Semestre de 2011

CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DISCIPLINA: ALGORITMOS e LÓGICA DE PROGRAMAÇÃO**SIGLA:****CARGA HORÁRIA SEMANAL:** 04**CARGA HORÁRIA SEMESTRAL:** 80 aulas**PROFESSOR:** Fernando Masanori Ashikaga**OBJETIVOS:** Analisar problemas computacionais e projetar soluções por meio da construção de algoritmos.**EMENTA:** Projeto e representação de algoritmos. Estruturas de controle de fluxo de execução: seqüência, seleção e repetição. Tipos de dados básicos e estruturados (vetores e registros). Rotinas. Arquivos. Implementação de algoritmos usando uma linguagem de programação.**METODOLOGIA:**

- Aulas expositivas com exemplos e comentários;
- Aulas práticas em laboratório;

CRITÉRIOS DE AVALIAÇÃO:

Serão feitas as seguintes avaliações:

- Provas: P_1 e P_2 (valem de 0 a 10)
- Exercícios Programa: EP_1 e EP_2 (também para os alunos de DP). Obs.: não faça sua implementação a partir de código alheio, isso configura plágio.
- Listas de exercícios. Não valem nota, mas é absolutamente necessário fazê-las.

Média de Provas : $P = (P_1 + 2 \cdot P_2) / 3$ Média de Exercícios Programa: $EP = (EP_1 + 2 \cdot EP_2) / 3$ Se $P \geq 6$ e $EP \geq 6$ (ambas as médias devem ser maiores que 6)

$$MF = (2 \cdot P + EP) / 3$$

caso contrário, a média final será

$$MF = \min \{EP, P\}$$

Haverá uma prova substitutiva no final do semestre, cujo conteúdo é a matéria toda.

As provas e exercícios programa são individuais. A constatação de plágio não somente implica nota zero na avaliação, como também a reprovação automática na disciplina. Aqui se inclui também o aluno que deixou copiar sua prova ou exercício programa.

Aula	Dia	Conteúdo Programático
Agosto		
1	02	Apresentação da disciplina. Motivação.
2	03	Variáveis e entrada de dados.
3	09	Condições.
4	10	Repetições.
5	16	Listas.
6	17	Strings.
7	23	Funções.
8	24	Arquivos.
9	30	Classes e Objetos.
10	31	Adivinhando números: atribuição, comparação, if/else, while, números randômicos.
Setembro		
11	03	Strings. Acessando páginas HTML. Teste de Mesa. Reposição no sábado.
12	06	Funções. Enviando mensagens ao Twitter.
13	13	Funções. Valores de retorno. Variáveis locais.
14	14	Arquivos e listas.
15	20	Dicionários e bancos de dados.
16	21	Dicionários e bancos de dados.
17	27	Programação modular.
18	28	Utilizando interfaces gráficas. Entrega EP₁.
Outubro		
19	01	Pygame. Reposição no sábado.
20	04	Semana de Provas.
21	05	Semana de Provas.
22	11	Exercícios com interfaces gráficas. Reposição no sábado.
23	18	Tratamento de Exceções.
24	19	Mixando áudio. Noções de objetos.
25	22	Mixando áudio. Noções de objetos. Reposição no sábado.
26	25	Google Python Class.
27	26	Google Python Class.
Novembro		
28	01	Google Python Class.
29	08	Semana da Tecnologia.
30	09	Google Python Class.
31	16	Projeto de algoritmos e recursão.
32	19	Projeto de algoritmos e recursão.
33	22	Programação Web com Web2py. Django se houver tempo.
34	23	Programação Web com Web2py. Django se houver tempo.
35	29	Programação nas nuvens utilizando o Google App Engine.
36	30	Programação nas nuvens utilizando o Google App Engine. Entrega do EP₂.
Dezembro		
37	06	Semana de Provas.
38	07	Semana de Provas.
39	13	Prova Substitutiva.
40	14	Plantão de dúvidas sobre as notas finais.

Bibliografia básica:

- MENEZES, N.N.C., Introdução à Programação com Python – Algoritmos e lógica de programação para iniciantes, Novatec, 2010, 222p. (Livro texto para as primeiras semanas, excelente para quem está começando a programar).
- GRIFFITHS, D., BARRY, P., Head First Programming – A learner's guide to programming using the Python language, O'Reilly, 2009, 406p. (Procura motivar iniciantes no aprendizado da programação utilizando exemplos interessantes como posts no Twitter, games e consulta à banco de dados. Existe a versão em português vendida pela Altabooks).
- BARRY, P., Head First Python, O'Reilly, 2010, 457p. (Aborda tópicos como programação em dispositivos móveis via Android ou Web utilizando o Google AppEngine).
- SANDERSON, D., Programming Google App Engine, O'Reilly, 2010, 367p. (Programação nas nuvens utilizando a plataforma da Google, códigos em Python e Java).
- Google Python Class, curso de Python ministrado na Google, 2010, <http://code.google.com/intl/en/edu/languages/google-python-class/>

Bibliografia complementar:

- ZELLE, J., Python Programming: An Introduction to Computer Science, Second Edition, Franklin, Beedle & Associates, 2010, 514p. (Estudo mais sistemático da programação).
- LUTZ, M., Learning Python, 4rd Ed, O'Reilly, 2009, 1216p. (Referência completa da linguagem Python, explica exaustivamente as partes básicas para iniciantes).
- SANTANA, O., GALESI, T., Python e Django – Desenvolvimento ágil de aplicações web, Novatec, 2010, 279p. (Introdução ao framework de desenvolvimento de aplicações web Django).
- KNUTH, D.E., The Art of Computer Programming, Volumes 1-3 Boxed Set 2nd Edition, Addison-Wesley, 1998, 896p. (Antigo, porém fantástico para aprofundamento).
- SKIENA, S.S., REVILLA, M.A., Programming Challenges, Springer, 2003, 359p. (Espetacular repositório de problemas de treinamento para maratonas de programação).
- MARTELLI, A., Python in a Nutshell, 2nd Ed, 2006, 712p.

RESPONSÁVEL PELA DISCIPLINA

__/__/__

RESPONSÁVEL PELO CURSO

__/__/__

APROVADO PELA COMISSÃO DE IMPLANTAÇÃO EM __/__/__