

PLANO DE ENSINO - 1º Semestre de 2013

CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DISCIPLINA: ALGORITMOS e LÓGICA DE PROGRAMAÇÃO (Tarde)**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.
- 15 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)

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

caso contrário, a média final será

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

Coding Dojo: aulas extras para treinar programação (3ª e 4ª 18h45-19h35).**Big Brother:** o aluno que tiver $P_1 \geq 7$ poderá adotar um aluno que tirou $P_1 \leq 6$, assim ele terá direito a $(P_2 - P_1)/4$ do outro aluno caso ele melhore sua nota.**Prova substitutiva:** no final do semestre, cujo conteúdo é a matéria toda.**ATENÇÃO:** 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
Fevereiro		
1	04	Apresentação da disciplina. Motivação.
2	05	Variáveis e entrada de dados.
3	18	Condições.
4	19	Repetições. Teste de Mesa.
5	25	Listas.
6	26	Strings.
Março		
7	04	Funções.
8	05	Arquivos.
9	11	Classes e Objetos.
10	12	Adivinhando números: atribuição, comparação, if/else, while, números randômicos.
11	25	Strings. Acessando páginas HTML.
12	26	Funções. Hackeando o Twitter.
Abril		
13	01	Funções. Valores de retorno. Variáveis locais.
14	02	Arquivos e listas.
15	08	Semana de Provas.
16	09	Semana de Provas.
17	15	Dicionários e bancos de dados.
18	16	Dicionários e bancos de dados.
19	22	Hackeando o Facebook.
20	23	Interface Gráfica. Entrega do EP₁.
21	29	Google Python Class.
22	30	MVC.
Maió		
23	06	Google Python Class.
24	07	Tratamento de Exceções.
25	13	Google Python Class.
26	14	DJ Mix.
27	20	Google Python Class.
28	21	Google Python Class.
29	27	Pygame.
30	28	Orientação à Objetos.
Junho		
31	03	Projeto de algoritmos e recursão.
32	04	Projeto de algoritmos e recursão.
33	10	Programação Web com Web2py.
34	11	Resolução de Exercícios.
35	17	Semana de Provas.
36	18	Semana de Provas
37	24	Plantão de dúvidas para o EP ₂ .
38	25	Entrega do EP₂.
39	01	Plantão de dúvidas (julho)
40	02	Prova Substitutiva (julho)

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 __/__/__