



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
PRÓ-REITORIA DE ENSINO DE GRADUAÇÃO

Rua Dom Manoel de Medeiros, s/n – Dois Irmãos 52171-900 Recife-PE
Fone: 0xx-81-332060-40 proreitor@preg.ufrpe.br

PLANO DE ENSINO

I – IDENTIFICAÇÃO

CURSO: Bacharelado em Sistemas de Informação

MODALIDADE: Presencial

DISCIPLINA: Introdução à programação (Turma SI2)

PRÉ-REQUISITO: não há

(X) OBRIGATÓRIA () OPTATIVA

DEPARTAMENTO: Departamento de Estatística e Informática

PROFESSOR RESPONSÁVEL: Marcelo Gama

ANO: 2013

SEMESTRE LETIVO: () Primeiro (x) Segundo

TOTAL DE CRÉDITOS: 4

CARGA HORÁRIA: 90 horas

II - EMENTA (Sinopse do Conteúdo)

Fundamentos da construção de algoritmos e programas. Ambientes de programação: uso de uma linguagem de programação. Conceitos básicos: variáveis, operadores e expressões, estruturas de controle (atribuição, seleção, repetição). Dados estruturados: listas, cadeias, dicionários, tuplas). Subprogramas: funções, procedimentos. Parâmetros locais e globais. Recursão. Ordenação interna: bubblesort, inserção, shellsort, heapsort, quicksort. Pesquisa interna: seqüencial, binária. Modularização. Complexidade temporal de algoritmos. Introdução a programação orientada a eventos. Introdução a programação orientada a objetos. Introdução a programação orientada a aspectos. Projeto: desenvolvimento de um programa de porte médio.

III - OBJETIVOS DA DISCIPLINA

GERAL

- Ajudar no desenvolvimento de um raciocínio lógico estruturado que facilitará o desenvolvimento de algoritmos para resolver problemas, independente de linguagem, que necessitem de programação.

ESPECÍFICOS

- Apresentar os principais conceitos de lógica de programação.
- Estudar o conceito e representação de algoritmos e aprender a utilizá-los na resolução de problemas.
- Estabelecer uma relação entre algoritmos e lógica de programação.
- Aprender a utilizar de forma correta os mais diversos tipos de estruturas de dados.
- Conhecer as principais estruturas de controles e sua utilização na construção de algoritmos.
- Aprender a manipular arquivos.
- Entender o conceito de modularização e como utilizá-lo para reduzir a complexidade de problemas.
- Conhecer os principais paradigmas de programação.
- Estudar o conceito de complexidade de algoritmos.
- Aplicar os conceitos aprendidos na disciplina desenvolvendo um aplicativo de porte médio.

IV - CONTEÚDO PROGRAMÁTICO

1. Algoritmos e linguagens

- 1.1** Algoritmos e fluxogramas.
- 1.2** Instruções de um algoritmo.
- 1.3** Tipos de linguagem (máquina, simbólica, abstrata).
- 1.4** Programa, compilação, interpretação e execução.

2. Ambiente de programação

- 2.1.** Conceitos básicos de uma linguagem: variáveis, tipos, estruturas de controle.
- 2.2.** Declaração de tipo simples.
- 2.3.** Comandos de entrada e saída.

- 2.4. Comando de atribuição.
- 2.5. Operadores e expressões.

3. Tipos estruturados

- 3.1. Listas.
- 3.2. Filas.
- 3.3. Array unidimensional (vetor).
- 3.4. Array bidimensional (matriz).
- 3.5. Arquivos.

4. Subprogramas

- 4.1 Função com parâmetro
- 4.2 Sub-rotinas com parâmetro.
- 4.3 Passagem de parâmetro.

5. Algoritmos

- 5.1 Pesquisa sequencial e binária.
- 5.2 Bubblesort.
- 5.3 Inserção.
- 5.4 Shellsort.
- 5.5 Heapsort.
- 5.6 quicksort.

6. Noções de complexidade e paradigmas de programação

- 6.1. Notação o , O , Ω e Θ .
- 6.2. Introdução a programação orientada a eventos.
- 6.3. Introdução a programação orientada a objetos.
- 6.4. Introdução a programação orientada a aspectos.

7. Projeto de implementação

V – MÉTODOS DIDÁTICOS DE ENSINO

- (x) Aula expositiva
- () Seminário
- (x) Leitura dirigida
- (x) Demonstração (prática realizada pelo professor)
- (x) Laboratório (prática realizada pelo aluno)
- () Trabalho de campo
- () Execução de pesquisa

I - CRITÉRIOS DE AVALIAÇÃO

1ª VA = Prova escrita: 6,0 pontos
Implementações em laboratório: 2,0 pontos
Listas de implementações: 2,0 pontos

2ª VA = Implementações em laboratório: 2,0 pontos
Listas de implementações: 2,0 pontos
Projeto de implementação: 6,0 pontos

3ª VA = Prova escrita: 10,0 pontos

Final = Prova escrita: 10,0 pontos

OBS1: A 3ª VA e a Final incluem todo o conteúdo do semestre.

OBS2.: A média de um aluno é calculada pela fórmula

$$\text{Média} = \frac{\text{Maior nota (VA1, VA2, VA3)} + \text{Segunda maior nota (VA1, VA2, VA3)}}{2}$$

Um aluno é considerado aprovado por média quando $\text{Média} \geq 7,0$. Caso contrário, ele terá que fazer a prova final e será aprovado quando

$$\text{Média final} = \frac{\text{Média} + \text{Nota da Final}}{2} \geq 5,0$$

OBS3: A 3ª. VA pode ser feita por quem não fez a prova escrita da 1ª. VA ou para melhorar a média.

OBS4: O aluno que não entregar o projeto de implementação referente à 2ª. VA não poderá usar a 3ª. VA para substituir essa nota e estará automaticamente reprovado.

CRONOGRAMA	
AULAS	CONTEÚDOS
	Introdução à programação
01.	Apresentação da disciplina. Computadores, fluxogramas, algoritmos e programas.
	Pseudocódigos e Visualg
02.	Pseudocódigos. Palavras reservadas. Tipos primitivos, variáveis e expressões. Blocos. Comandos de entrada e saída. Ferramenta Visualg.
03.	Estruturas de controle.
04.	Estruturas de controle. Recursão.
05. ★	Prática em laboratório. Entrega da lista 1.
06.	Estruturas de dados: Arrays (vetores e matrizes) e listas.
07.	Estruturas de dados: Filas, pilhas. Registros e arquivos.
08. ★	Prática em laboratório. Entrega da lista 2.
	A linguagem de programação Python
09.	Instalação. Ambiente de programação. Compilação e execução. Tipos primitivos de dados. Formatação de strings. Entrada e saída de dados.
10.	Estruturas de controle.
11.	Dicionários. Tratamento de erros.
12. ★	Prática em laboratório. Entrega da lista 3.
13.	Modularização: Função com parâmetros, sub-rotina com parâmetros, passagem de parâmetros.
14.	Estruturas de dados.
15. ★	Prática em laboratório. Entrega da lista 4.
16.	Aula de revisão.
17. ★	Primeira VA.
	Projeto: Especificações
18.	Apresentação de propostas de projeto. Discussão dos requisitos mínimos exigidos. Apresentação do cronograma de execução.
	Programação orientada a objetos com Java
19.	Fundamentos da programação orientada a objetos: Classes, objetos, herança e polimorfismo. Introdução à linguagem de programação Java.
20.	Classes em Java. Propriedades e métodos. Classes derivadas, abstratas e finais. Interfaces.
21.	Tratamento de exceções. Definindo e gerando exceções. Uso das classes da API padrão de Java.
22. ★	Prática em laboratório. Entrega da lista 5.
	Projeto: Checkpoint 1
23.	Apresentação do andamento dos projetos.

CRONOGRAMA – CONTINUAÇÃO

AULAS	CONTEÚDOS
	Outros paradigmas de programação
24.	Noções de programação orientada a eventos. Noções de programação orientada a aspectos.
	Algoritmos
25.	Algoritmos de ordenação: Pesquisa sequencial e binária. Bubblesort. Inserção.
26.	Algoritmos de ordenação: Shellsort. Heapsort. Quicksort.
27.	Noções de algoritmos aleatorizados. Testes de primalidade. Teste de igualdades polinomiais. Teste de igualdades de strings longas e cadeias de DNA.
28. ★	Prática em laboratório. Entrega da lista 6.
	Projeto: Checkpoint 2
29.	Apresentação do andamento dos projetos.
	Projeto: Finalização
30.	Verificação do andamento dos projetos e últimos ajustes. Sorteio da ordem de apresentação.
	Projeto: Apresentação dos projetos
31. ★	Apresentação de projetos.
32. ★	Apresentação de projetos.
	Avaliações finais
33. ★	3a. VA
34. ★	Avaliação final

VIII – BIBLIOGRAFIA (Conforme normas da ABNT)

BÁSICA

1. Forbellone, André Luiz Villar; Eberspacher, Henri Frederico; Lógica de Programação - A Construção de Algoritmos e Estruturas de Dados. 3ª Edição. Makron Books, 2007.
2. Lopes, Anita e Garcia, Guto. Introdução a Programação. Editora Campus, 2002.
3. Cormen, Thomas H. et. al. Algoritmos: Teoria e Prática. Editora Campus, 2002.

COMPLEMENTAR:

1. Ziviani, Nivio. Projeto de Algoritmos. Editora Nova Fronteira, 2004.
2. Sebesta, Robert W. Conceitos de Linguagens de Programação. Bookman, 2001.
3. Nieto, T. R. Internet & World Wide Web. Como Programar. Bookman, 2003.
4. Deitel, Harvey M. et. al. XML Como Programar. Bookman, 2003.
5. Van Rossum, Guido. Tutorial de Python. Disponível gratuitamente em <http://python.org/>, 2004.
6. Horstmann, Cay, et al. Core Java 2, Volume 1: Fundamentos. Makron Books,. São Paulo, 2001

Recife, _____ de _____ de _____

Professor Responsável