



Plano de Ensino

1. Dados de Identificação

Componente Curricular: Algoritmo e Programação

Curso: Ciências da Computação

Período: Primeiro

Ano/Semestre: 2011/2

Carga Horária/Créditos: 60 horas (72 horas/aula) / 4

Professor: Dr. Rafael Piccin Torchelsen e Ms. Fernando Belvilacqua

Monitor:

2. Ementa

Conceito e construção de algoritmos. Tipos básicos de dados. Comandos de atribuição, condicionais e de repetição. Registros. Vetores e Matrizes. Modularização.

3. Justificativa

As disciplinas de programação são bases teóricas para várias outras disciplinas do curso de computação além de proporcionar ao acadêmico conhecimentos necessários para modelar problemas reais para serem resolvidos por computadores.

4. Objetivo

Desenvolver o raciocínio lógico para o desenvolvimento de algoritmos. Compreender como o computador executa programas. Desenvolver algoritmos simples utilizando os conceitos aprendidos.

4. Conteúdo Programático

Aulas	Total Parc.		Assunto	Capítulo do Livro Texto
2	2	3/8	Introdução à disciplina; apresentação do plano de ensino; estrutura lógica dos computadores	Cap. 1
2	4	5/8	Introdução à disciplina; apresentação do plano de ensino; estrutura lógica dos computadores	Cap. 1
2	6	10/8	Algoritmos, fluxograma, pseudo-código, linguagens de programação, compiladores	Cap. 2
2	8	12/8	Algoritmos com entrada, saída e processamento	Cap. 2
2	10	17/8	Introdução à linguagem de programação C (código-fonte, compilador, arquivo .c, printf(), IDE)	Cap. 2
2	12	19/8	Introdução a variáveis (memória, sintaxe, tipos, atribuição); básico sobre printf()	Cap. 3, 4 e 5
2	14	24/8	Operações aritméticas com variáveis; básico sobre scanf()	Cap. 4 e 5
2	16	26/8	Condições simples e operadores de comparação (==, !=, >=, <=, etc)	Cap. 6
2	18	31/8	Explicação sobre if, if..else, if..else if, else	Cap. 6
2	20	2/9	Condições compostas e tabela verdade	Cap. 6
2	22	9/2	Condições compostas com parênteses; procedência de operadores	Cap. 6
2	24	14/9	Estruturas de repetição (while e do...while) e iterações	Cap. 6
2	26	16/9	Estruturas de repetição (for)	Cap. 6
2	28	21/9	Exercícios	
2	30	23/9	Exercícios	
2	32	28/9	Primeira prova (P1), conteúdo: Tudo visto até o dia da prova	
2	34	30/9	Modularização e funções (protótipo, parâmetros, retorno, escopo)	Cap. 8
2	36	5/10	Funções (escopo)	Cap. 8
2	38	7/10	Organização de programa em arquivos .c e .h; bibliotecas disponíveis	
2	40	14/10	Recuperação NP1, conteúdo: mesmo de P1	
2	42	19/10	Introdução a vetores	Cap. 7
2	44	21/10	Aprofundamento sobre vetores (iterações, etc)	Cap. 7
2	46	26/10	Exercícios	
2	48	4/11	Vetores bidimensionais (matrizes)	Cap. 7

Aulas	Total Parc.		Assunto	Capítulo do Livro Texto
2	50	9/11	Aprofundamento sobre matrizes	Cap. 7
2	52	11/11	Introdução a structs	Cap. 10
2	54	16/11	Aprofundamento sobre structs	Cap. 10
2	56	18/11	Exemplos de uso sobre structs	Cap. 10
2	58	23/11	Exercícios	
2	60	25/11	Ponteiros	
2	62	30/11	Ponteiros	
2	64	2/12	Segunda prova (P2), conteúdo: Tudo visto até o dia da prova	
2	66	7/12	Apresentação do trabalho final	
2	68	9/12	Recuperação NP2, conteúdo: Tudo visto até o dia da prova	
2	70	14/12	Apresentação do trabalho final	
2	72	16/12	Exame	

* O plano e cronograma podem ser alterados pelo professor ao longo do semestre. O aluno deve acompanhar o cronograma atualizado através do Moodle.

6. Estratégias de ensino

Conduzir a disciplina com aulas expositivas/dialogadas enquanto discutidos os itens de cunho teórico, evoluindo em tópicos específicos para exercícios práticos, demonstrações, contextualização baseada em publicações atualizadas. Uso de atividades em laboratórios com o objetivo de apresentar/exercitar os conceitos estudados.

7. Avaliação

Uso de abordagens tais como: provas teóricas, avaliação escrita em aula, exercícios extra-classe, trabalhos de implementação, entre outros.

As avaliações serão agrupadas em dois momentos (conforme instrução normativa No. 001/Prograd/2010) Notas Parciais 1 e 2 (NP1 e NP2, respectivamente). A NP1 será composta por uma avaliação escrita (P1) e trabalhos extra-classe (G1) realizados até a data da prova, com o seguinte cálculo:

$$NP1 = P1 + G1$$

$$\text{onde } P1 \geq 0 \text{ e } P1 \leq 7 \text{ e } G1 \geq 0 \text{ e } G1 \leq 3$$

sendo G1 calculado da seguinte forma:

$$G1 = T_1 + T_2 + \dots + T_n$$

onde T_i representa a nota de um trabalho, onde o peso e a quantidade de trabalhos são decididos em aula.

A NP2 será composta por uma avaliação escrita (P2) e trabalhos extra-classe (G2) realizados até a data da prova, com o seguinte cálculo:

$$NP2 = P2 + G2$$

$$\text{onde } P2 \geq 0 \text{ e } P2 \leq 7 \text{ e } G2 \geq 0 \text{ e } G2 \leq 3$$



sendo G2 calculado da seguinte forma:

$$G2 = T_1 + T_2 + \dots T_n$$

onde T_i representa a nota de um trabalho, onde o peso e a quantidade de trabalhos são decididos em aula.

A média final (MF) será calculada como $MF=(NP1+NP2)/2$

Para cada NP será ofertada prova de recuperação (RP). A reposição de nota se aplica somente à prova, **não substituindo os trabalhos**. Além disso, RP não substitui P, mas sim é feita uma **média** entre RP e P. Dessa forma, para os alunos que prestarem RP o cálculo de NP é definido por:

$$NP1 = ((P1+RP1)/2) + G1$$

e

$$NP2 = ((P2+RP2)/2) + G2$$

Durante os 5 minutos iniciais de RP o aluno terá a oportunidade de avaliar à prova e decidir entre prestar ou não a mesma. Para os que decidirem por não prestar RP o cálculo de NP não é alterado.

Aos estudantes que não obtiverem média final igual ou superior a 6 será oferecido um exame final (EF). A nota do EF substituirá as notas de todas as provas (note que as notas dos trabalhos não são afetadas) e o cálculo da NP1 e NP2 será recalculada. A média final mínima de 6 pontos mantém-se para fins de aprovação.

Em caso de plágio as seguintes regras serão aplicadas.

Prova:

–O aluno recebe nota zero na prova onde o plágio for detectado

Trabalhos:

–É permitido usar conteúdo da internet, livros, colegas, etc., contanto que uma citação seja feita. A nota do trabalho será proporcional ao conteúdo original.

–Caso seja detectado plágio o aluno recebe zero no trabalho em questão, além disso, por demonstrar prática não aceitável o caso será levado ao conhecimento do colegiado.

8. Atendimento ao aluno

Horário: Quarta-feira entre 16:00 e 20:00

Local: Sala dos professores

Fora desse horário somente com agendamento através do email: rafael.torchelsen@uffs.edu.br

9. Referências

REFERÊNCIAS BÁSICAS

LOPES, A., GARCIA, G. Introdução à Programação. Rio de Janeiro: Campus, 2002.

VILARIM, G. Algoritmos: Programação para Iniciantes. Rio de Janeiro: Ciência Moderna, 2004.

FORBELLONE, A. Lógica de Programação: A Construção de Algoritmos e Estruturas de Dados. São Paulo: Makron Books, 2005.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo. Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. 19 ed.: Érica, 2001.

REFERÊNCIAS COMPLEMENTARES

LOPES, A.; GARCIA, G. **Introdução à Programação: 500 algoritmos resolvidos**. Campus, 2002.

ARAÚJO, E. C. **Algoritmos: Fundamentos e Prática**. Florianópolis: Visual Books, 2ed., 2005.

BORATTI, I. C. **Introdução à Programação: Algoritmos**. Florianópolis: Visual Books, 2ed., 2004

Textos disponíveis no Moodle:



Universidade Federal da Fronteira Sul

Apostila da UFMG

Apostila da UFRJ

Evaristo, Jaime. Aprendendo a Programar Programando na Linguagem C Para Iniciantes, versão em PDF que se encontra no Moddle.