

Plano de Ensino

1. Dados de Identificação

Componente Curricular: Algoritmo e Programação
Curso: Ciência da Computação
Período: Primeiro
Ano/Semestre: 2013/2
Carga Horária/Créditos: 60 horas (72 horas/aula) / 4
Professor: Dr. Rafael Piccin Torchelsen

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.

5. Conteúdo Programático

Total Parc.	Assunto
2	Introdução à disciplina; apresentação do plano de ensino; estrutura lógica dos computadores
5	Algoritmos, fluxograma, pseudo-código, linguagens de programação, compiladores
7	Algoritmos com entrada, saída e processamento
10	Introdução à linguagem de programação C (código-fonte, compilador, arquivo .c, printf(), IDE)
12	Introdução a variáveis (memória, sintaxe, tipos, atribuição); básico sobre printf()
15	Operações aritméticas com variáveis; básico sobre scanf()
17	Condições simples e operadores de comparação (==, !=, >=, <=, etc)
20	Explicação sobre if, if..else, if..else if, else
22	Condições compostas e tabela verdade; Condições compostas com parênteses; Procedência de operadores;
25	Estruturas de repetição (while e do...while) e iterações
27	Estruturas de repetição (for)
30	Exercícios

32	Exercícios
35	Primeira prova (P1), conteúdo: Tudo visto até o dia da prova
37	Modularização e funções (protótipo, parâmetros, retorno, escopo)
40	Recuperação NP1, conteúdo: mesmo de P1
42	Bibliotecas de funções
45	Introdução a vetores
47	Aprofundamento sobre vetores
50	Vetores bidimensionais (matrizes)
52	Aprofundamento sobre matrizes
55	Introdução a structs
57	Aprofundamento sobre structs
60	Ponteiros
62	Ponteiros
65	Segunda prova (P2), conteúdo: Tudo visto até o dia da prova
67	Apresentação do trabalho final
70	Recuperação NP2, conteúdo: mesmo de P2
72	Apresentação do trabalho final

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 e práticas, avaliação escrita em aula, exercícios extraclasse, 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 e prática (P1), trabalhos extraclasse (G1) realizados até a data da prova, com o seguinte cálculo:

$$NP1 = P1 + G1$$

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

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 e prática (P2), trabalhos extraclasse (G2) realizados até a data da prova, com o seguinte cálculo:

$$NP2 = P2 + G2$$

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

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 uma 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 quando a nota de P for superior a nota de RP. Quando a nota obtida na RP for inferior que P a nota de RP é desconsiderada. Dessa forma, para os alunos que prestarem RP e obtiverem nota superior a P 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 a prova e decidir entre prestar ou não a mesma. Para os que decidirem por não prestar RP o cálculo de NP não é alterado.

Em caso de plágio as seguintes regras serão aplicadas.

Prova:

-O aluno recebe nota zero na prova onde o plágio foi 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: Quinta-feira entre 18:00 e 19:00

Local: Sala dos professores

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

9. Referências

REFERÊNCIAS BÁSICAS

1. LOPES, A., GARCIA, G. Introdução à Programação. Rio de Janeiro: Campus, 2002.
2. VILARIM, G. Algoritmos: Programação para Iniciantes. Rio de Janeiro: Ciência Moderna, 2004.
3. FORBELLONE, A. Lógica de Programação: A Construção de Algoritmos e Estruturas de Dados. São Paulo: Makron Books, 2005.
4. MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo. Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. 19 ed.: Érica, 2001.

REFERÊNCIAS COMPLEMENTARES



1. LOPES, A.; GARCIA, G. Introdução à Programação: 500 algoritmos resolvidos. Campus, 2002.
2. ARAÚJO, E. C. Algoritmos: Fundamentos e Prática. Florianópolis: Visual Books, 2ed., 2005.
3. BORATTI, I. C. Introdução à Programação: Algoritmos. Florianópolis: Visual Books, 2ed., 2004