

Curso:	Engenharias	Semestre/Ano:	02/2018
Disciplina:	Algoritmos e Programação de Computadores	Créditos:	06
Carga Horária:	90 horas	Turma:	CC
Professor:	Matheus de Sousa Faria		

Plano de Ensino

1 Objetivo da Disciplina

A disciplina Algoritmos e Programação de Computadores (APC) busca propiciar aos alunos conhecimentos de lógica de programação e algoritmos. Nela serão trabalhados o raciocínio lógico e a capacidade de resolver problemas através do uso da programação. E apresentará como desenvolver e testar programas para estes problemas.

2 Ementa do Programa

- | | |
|---|---|
| I. Princípios fundamentais de construção de programas | XI. Entrada e saída de dados |
| II. Construção de algoritmos e sua representação em pseudocódigo e linguagens de alto nível | XII. Estruturas de dados estáticas: agregados homogêneos e heterogêneos |
| III. Noções de abstração | XIII. Iteração e recursão |
| IV. Especificação de variáveis e funções | XIV. Noções de análise de custo e complexidade |
| V. Testes e depuração | XV. Desenvolvimento sistemático e implementação de programas |
| VI. Padrões de soluções em programação | XVI. Estruturação, depuração, testes e documentação de programas |
| VII. Noções de programação estruturada | XVII. Resolução de problemas |
| VIII. Identificadores e tipos | XVIII. Aplicações em casos reais e questões ambientais |
| IX. Operadores e expressões | |
| X. Estruturas de controle: condicional e repetição | |

3 Horário das Aulas e Atendimento

Aulas: segundas das 14:00 às 15:50, quartas e sextas, das 10:00 às 11:50 hrs.

Atendimento: quartas e sextas, das 08:00 às 10:00 hrs.

4 Metodologia

A metodologia consiste em aulas expositivas, com o auxílio do quadro branco e projetor digital. A fim de fortalecer a aprendizagem da disciplina, as aulas serão complementadas com exercícios e atividades, presenciais e extra-classe.

5 Critérios de Avaliação

A avaliação do curso será feita através de 4 (quatro) provas práticas e listas de exercícios.

5.1 Provas

Cada prova é composta de problemas relacionados com o conteúdo apresentado até a data da prova. Cada problema será corrigido segundo os mesmos critérios de correção aplicados em competições de programação: após ser compilado de forma bem sucedida, uma série de testes unitários automatizados alimentarão o programa com entradas válidas, e comparará os resultados obtidos com as saídas corretas. Um problema será considerado correto se obtiver sucesso em todos os testes unitários.

O aluno só receberá a pontuação devida ao problema caso o problema esteja correto. Cada prova P_i será pontuada em um total de 10 pontos.

No final do semestre será aplicada uma prova substitutiva, cujo resultado substituirá o último pior resultado dentre os obtidos nas quatro provas práticas, independentemente do resultado da prova substitutiva. Todos os alunos podem fazer a prova substitutiva, se assim desejarem.

5.2 Listas

As listas de exercícios serão compostas por uma série de problemas relacionados aos tópicos da ementa do curso. O aluno deverá submeter as soluções destes problemas via plataforma URI (urionlinejudge.com.br), em procedimento a ser detalhado ao longo do curso.

Haverão listas de exercícios semanalmente, nas semanas onde possuem conteúdos práticos. Cada lista de exercício terá o tempo de uma semana para ser entregue via URI.

5.3 Menção Final

A nota final (N_F) é composta pela nota de provas, sendo calculada da seguinte maneira.

$$N_F = \frac{P_1 + 2 * P_2 + 3 * P_3 + 4 * P_4}{10}$$

E a menção final é dada pela tabela seguinte:

Menção	Descrição	Nota
SR	Sem Rendimento	0
II	Inferior	de 0,1 a 2,9
MI	Médio Inferior	de 3,0 a 4,9
MM	Médio	de 5,0 a 6,9
MS	Médio Superior	de 7,0 a 8,9
SS	Superior	de 9,0 ou maior

5.4 Critérios de Aprovação

Obterá aprovação no curso o aluno que cumprir as duas exigências abaixo:

1. Ter presença em 75% ou mais das aulas ministradas
2. Obter menção final igual ou superior a MM

6 Cronograma

Datas	Conteúdo	Prazo da Lista
13/08	Apresentação do Plano de Ensino. Ambiente de Desenvolvimento	
15/08	Comandos Básicos do Linux. Testando seu ambiente	
17/08	Introdução a Algoritmos. Ferramentas de trabalho	
20/08	Linguagem C: Tipos Primitivos, variáveis, operadores aritméticos	
22/08	Entrada e Saída	
24/08	Prática com o URI	
27/08	Estruturas Condicionais: if-else. Operadores Lógicos e Relacionais	Lista 1
29/08	Estruturas Condicionais: if-else	
31/08	Estruturas Condicionais: if-else	

03/09	Estruturas Condicionais: switch	Lista 2
05/09	Estruturas Condicionais: switch	
07/09	FERIADO	
10/09	Revisão para a prova	Lista 3
12/09	Prova 1	
14/09	Resolução da Prova 1. Recaptulação.	
17/09	Estruturas de Repetição: while, do-while	
19/09	Estruturas de Repetição: while, do-while	
21/09	Estruturas de Repetição: while, do-while	
24/09	Semana Universitária (Não haverá aula)	Lista 4
26/09	Semana Universitária (Não haverá aula)	
28/09	Semana Universitária (Não haverá aula)	
01/10	Estruturas de Repetição: for / Funções	Lista 5
03/10	Estruturas de Repetição: for / Funções	
05/10	Estruturas de Repetição: for / Funções	
08/10	Revisão para a prova	Lista 6
10/10	Prova 2	
12/10	FERIADO	
15/10	Arrays	
17/10	Arrays	
19/10	Arrays	
22/10	Matrizes	Lista 7
24/10	Matrizes	
26/10	Matrizes	
29/10	Strings	Lista 8
31/10	Strings	
02/11	FERIADO	
05/11	Modularização e Structs	Lista 9
07/11	Modularização e Structs	
09/11	Modularização e Structs	
12/11	Revisão para a prova	Lista 10
14/11	Prova 3	
16/11	Resolução da Prova 3. Recaptulação	
19/11	Recursão	

21/11	Recursão	
23/11	Recursão	
26/11	Revisão para prova	Lista 11
28/11	Prova 4	
30/11	Revisão de Notas	
03/12	Revisão para prova substitutiva	Lista 12
05/12	Prova Substitutiva	
07/12	Revisão de Notas	

7 Bibliografia

Básica

Cormen, T. et al., Algoritmos: Teoria e Prática. 3a ed., Elsevier - Campus, Rio de Janeiro, 2012
 Ziviani, N., Projeto de Algoritmos com implementações em Pascal e C, 3a ed., Cengage Learning, 2010.

Felleisen, M. et al., How to design programs: an introduction to computing and programming, MIT Press, EUA, 2001.

Complementar

Evans, D., Introduction to Computing: explorations in Language, Logic, and Machines, CreatSpace, 2011.

Harel, D., Algorithmics: the spirit of computing, Addison-Wesley, 1978.

Manber, U., Introduction to algorithms: a creative approach, Addison-Wesley, 1989.

Kernighan, Brian W; Ritchie, Dennis M., C, a linguagem de programação: Padrão ansi. Rio de Janeiro: Campus

Farrer, Harry. Programação estruturada de computadores: algoritmos estruturados. Rio de Janeiro: Guanabara Dois, 2002.