

## **Plano de Ensino**

### **5LIEE**

### **Linguagem de Programação**

**60 horas**

### **Ementa**

Ao concluir esta disciplina, o aluno será capaz de solucionar problemas do mundo real através da criação de estratégias básicas de lógica de programação; compreender a formulação geral e a interação entre uma linguagem de programação estruturada de alto nível e as modernas arquiteturas de computadores; utilizar um compilador de linguagem de programação e uma interface de desenvolvimento para produzir novas soluções de programas de computador, adequando e otimizando a solução do problema; interpretar e verificar a corretude, bem como propor reparos a códigos de programas de computador já confeccionados; otimizar códigos de programa de computador já produzidos em uma linguagem de computador.

### **Objetivos**

- 1 - Solucionar problemas do mundo real através da criação de estratégias básicas de lógica de programação.
- 2 - Criar códigos que considerem a integração entre estruturas de programação de alto nível e as modernas arquiteturas de computadores.
- 3 - Produzir novas soluções de programas de computador, adequando e otimizando a solução do problema, através do uso de um compilador e uma interface de desenvolvimento.
- 4 - Identificar problemas, reparar ou corrigir formalmente códigos de programas de computador já em produção.
- 5 - Otimizar códigos de programa de computador produzidos por outrém em uma linguagem de computador estruturada de alto nível.

### **Conteúdos**

- 1 - Revisão de lógica de programação. Arquiteturas de computadores. Algoritmos. Execução de um algoritmo. Princípios de otimização de algoritmos. Desvios condicionais. Se-Então-Senão. Laços. Para; Enquanto-Faça; Repita-Até.
- 2 - Conceitos básicos de uma linguagem de programação. Estruturas lineares de linguagens de programação de alto nível estruturada. Compiladores. Interface de desenvolvimento. Processo de escrita-compilação-linking-execução.
- 3 - Conceitos básicos de C. Histórico da linguagem C. Indicações de uso. C na engenharia. Primeiros passos para implementação de algoritmos em pseudo-código para C. Exemplos de códigos migrados de pseudo-código para C.
- 4 - Uso básico de uma interface de programação em C. O compilador C. Sintaxe básica. Mensagens comuns do compilador. Primeiras mensagens de saída do compilador. Tratamento de erros. Identificação de código incorreto.
- 5 - Tipos de dados em C. Limites e variantes de dados em C. Uso de memória e otimização de dados em C. Comandos de entrada e saída de dados em C. Meu primeiro código em C (olá mundo). Validação de entrada e formatação de saída.
- 6 - Operadores aritméticos em C. Expressões aritméticas. Operadores relacionais em C. Operadores lógicos em C. Precedência de operadores em C. Operadores especiais em C (memória). Primeiros exercícios de programação em C. Integrando tudo junto.
- 7 - Vetores e matrizes em C. Alocação de memória de variáveis homogêneas multidimensionais. Laços para percorrer dados em vetores e matrizes. Acesso, alteração e movimentação de dados em vetores e matrizes em C.
- 8 - Uso de bibliotecas em C. Definição e papel da Biblioteca Padrão do C. Criação de bibliotecas próprias de código. Aspectos de reuso de códigos em C. Modularização de códigos em C. Diretrizes de compilação.
- 9 - Alocação dinâmica de memória em C. Conceito de endereçamento de memória. Uso de memória restrita ao programa. Aterramento de memória. Operadores de referenciamento de memória. Liberação de memória alocada dinamicamente.
- 10 - Uso de ponteiros para otimização de códigos. Vetores e matrizes operados por ponteiros. Ponteiros de ponteiros. Tipos de dados e ponteiros de tipos. Ponteiros sem tipo. Ponteiros de ponteiros. Problemas comuns de uso dos ponteiros
- 11 - Cadeias de caracteres em C. Conceitos de caracteres (strings). Abordagem de armazenamento de

caracteres em C e seu paralelo com o tipo inteiro. Alocação dinâmica de ponteiros de cadeias de caracteres. Comparações de strings. Biblioteca de strings.

12 - Aspectos avançados de alocação dinâmica de memória e cadeias de caracteres. Movimentação de dados brutos de memória. Funções de manipulação de memória. Funções de manipulação de cadeias de caracteres da biblioteca padrão C.

13 - Arquivos em C. Operações com arquivos. Alocação, exclusão e abordagens de gravação de dados em arquivos em C. Correlação das funções de exibição e de gravação de dados em arquivos, em C. Funções de manipulação de arquivos em C.

14 - Um sistema mais complexo em C. Abordagem conjunta de agenda pessoal, com alocação dinâmica de memória, otimização, gravação de dados em arquivo. Reavendo dados do disco e carregando na memória. Técnicas básicas de busca e ordenação.

## **Bibliografia**

### **Básica**

1 - SOARES, Márcio Vieira; SOUZA, Marco Antonio Furlan de; GOMES, Marcelo Marques. Algoritmos e lógica de programação: um texto introdutório para engenharia. São Paulo: Cengage Learning, 2013.

2 - DEITEL, Paul J. C++ como programar. Porto Alegre: Bookman, 2004.

3 - SINTES, Anthony. Aprenda programação orientada a objetos em 21 dias. São Paulo: Pearson Education, 2002.

4 - NOOMAN, Robert; TUCKER, Allen B. Linguagens de programação - princípios e paradigmas. São Paulo: Mcgraw-hill Interamericana, 2009.

### **Complementar**

1 - KERNIGHAN, Brian W. C: a linguagem de programação. Rio de Janeiro: Livros Técnicos e Científicos, 1986.

2 - PAIVA, Severino. Introdução a ação: do algoritmo as linguagens atuais- inclui linguagem Java, PHP, ASP e Objeto Pascal. Rio de Janeiro: Ciência Moderna, 2008.program

3 - CAMPOS, Edilene Aparecida V. de; ASCENCIO, Ana Fernanda Gomes. Fundamentos da programação de computadores: algoritmos, pascal, C/C ++ e Java. São Paulo: Pearson Prentice Hall, 2007.

4 - MANZANO, José Augusto N. G. Algoritmos: lógica para desenvolvimento de programações de computadores. São Paulo: Erica, 2013.

5 - ASCENCIO, Ana Fernanda Gomes. Fundamentos de programação de computadores. São Paulo: Pearson, 2013.

6 - DEITEL, H. M.; DEITEL, P. J. Visual basic. NET como programar. São Paulo: Makron Books, 2003.

7 - MALDONADO, José Carlos. Introdução ao teste de software. Rio de Janeiro: Campus, 2007.