

MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA
DIRETORIA DE ENSINO (DIREN)
DEPARTAMENTO DE ENSINO SUPERIOR (DEPES)
DEPARTAMENTO DE INFORMÁTICA (DEPIN)
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET (CST-SI)

DEPARTAMENTO	PLANO DE CURSO DA DISCIPLINA
DEPIN - Departamento Acadêmico de Informática	ARQUITETURA DE LINGUAGENS DE PROGRAMAÇÃO

CÓDIGO	PERÍODO	ANO	SEMESTRE	PRÉ-REQUISITOS
GTSI 1409	Opt	2014	1	GTSI1422 Estruturas de Dados
CRÉDITOS	AULAS/SEMANA			
4	TEÓRICA	PRÁTICA	ESTÁGIO	
	4	0	0	TOTAL DE AULAS NO SEMESTRE
				72

EMENTA

Características importantes de linguagens de programação. Amarrações. Valores e tipos de dados. Expressões e comandos. Modularização. Polimorfismo. Tratamento de exceções. Alocação dinâmica de memória. Visão geral de linguagens funcionais e lógicas.

BIBLIOGRAFIA

Bibliografia básica

1. SEBESTA, R. W. *Conceitos de Linguagens de Programação*. 5ª edição. Editora Bookman.
2. VAREJÃO, F. *Linguagens de Programação: Conceitos e Técnicas*. Editora Campus/Elsevier.
3. GHEZZI, C., JAZAYERI, M. *Programming Language Concepts*. 3ª edição. John Wiley & Sons.

Bibliografia complementar

1. PRATT, T. W., ZELKOWITZ, M. V. *Programming Languages: Design and Implementation*. Prentice Hall
2. MITCHELL, J. C. *Foundations for Programming Languages*. MIT Press.
3. BLACKBURN, Patrick; BOS, Johan; and STRIEGNITZ, Kristina. *Learn Prolog Now*. London: King's College Publications. Disponível em <http://www.learnprolognow.org/lpnpage.php?pageid=online>.
4. O'SULLIVAN, Bryan; STEWART, Don, and GOERZEN, John. *Real World Haskell*. O'Reilly, ISBN . Disponível em <http://book.realworldhaskell.org/read/>.
5. SEIBEL, Peter. *Practical Common Lisp*. Apress, Berkely, CA, USA. Disponível em <http://www.gigamonkeys.com/book/>.

OBJETIVO GERAL

Apresentar ao aluno conceitos da arquitetura de linguagens de programação, segundo seus diversos paradigmas.

METODOLOGIA

- Aulas expositivas, contando com recursos audiovisuais.
- Aulas eventuais em laboratório de informática, com o uso de um sistema gerenciador de banco de dados relacional de mercado.
- Resolução de exercícios de fixação e propostos.

CRITÉRIO DE AVALIAÇÃO

A avaliação semestral envolve duas provas escritas (P1 e P2). As datas das provas são agendadas entre o professor e a turma. A média parcial (MP) será calculada pelo cômputo da média aritmética simples entre a nota P1 e P2:

$$MP = (P1 + P2) / 2$$

O aluno que faltar a uma das duas provas terá direito a uma avaliação alternativa, denominada segunda chamada, versando sobre todos os tópicos abordados no curso, e cuja data também é agendada entre docente e discentes. A nota obtida nessa 2ª chamada substituirá a da avaliação P1 ou P2 onde o aluno não esteve presente. Caso ele falte às duas avaliações, terá atribuído o grau ZERO em uma delas.

Opcionalmente o docente pode propor um trabalho prático em cada uma das avaliações, com vistas à composição das notas P1 e P2.

Segundo o regimento do CEFET-RJ, caso o aluno obtenha média parcial inferior a 3,0 (três e zero) estará reprovado diretamente. Graus MP maiores ou iguais a 7,0 (sete e zero) aprovam diretamente o aluno. Em situações onde o aluno tenha grau MP entre 3,0 inclusive e 7,0 exclusive, terá direito a uma prova final (PF), que, juntamente com a média parcial gerará uma nova média, denominada média final (MF). Essa média é calculada da seguinte forma:

$$MF = (MP + PF) / 2$$

Para ser aprovado, o aluno deve alcançar uma média final MF maior ou igual a 5,0 (cinco e zero). Caso contrário, estará reprovado, devendo repetir a componente curricular.

PROGRAMA

1. Características importantes de linguagens de programação
 - 1.1. Propriedades
 - 1.2. Especificação
 - 1.3. Métodos de implementação
 - 1.4. Paradigmas
 - 1.5. Evolução
2. Amarrações
 - 2.1. Tempos de amarração
 - 2.2. Identificadores
 - 2.3. Ambientes de amarração
 - 2.4. Definições e declarações
3. Valores e tipos de dados

- 3.1. Tipos primitivos
- 3.2. Tipos compostos
- 4. Expressões e comandos
 - 4.1. Tipos e avaliações de expressões
 - 4.2. Tipos de comandos
- 5. Modularização
 - 5.1. Abstrações
 - 5.2. Técnicas de modularização
 - 5.3. Compilação separada
 - 5.4. Gerência de memória
- 6. Polimorfismo
 - 6.1. Sistemas de tipos
 - 6.2. Tipos de polimorfismo
- 7. Tratamento de exceções
 - 7.1. Mecanismos de tratamento de exceções
- 8. Alocação dinâmica de memória
 - 8.1. Gerência de heap
 - 8.2. Coleta de lixo
- 9. Visão geral de linguagens funcionais e lógicas
 - 9.1. Linguagens funcionais: LISP, COMMON LISP, Haskell
 - 9.2. Linguagens lógicas: PROLOG