



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE - CTS  
DEPARTAMENTO DE COMPUTAÇÃO – DEC

PLANO DE ENSINO

SEMESTRE 2020.1

**I. IDENTIFICAÇÃO DA DISCIPLINA:**

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
DEC7565	Construção de Compiladores	2	2	72

HORÁRIO		MODALIDADE
TURMAS TEÓRICAS	TURMAS PRÁTICAS	Remota Assíncrona e Síncrona
3.1420-2	5.1420-2	

**II. PROFESSOR(ES) MINISTRANTE(S)**

Prof. Rodrigo Pereira

E-mail: [pereira.rodrico@posgrad.ufsc.br](mailto:pereira.rodrico@posgrad.ufsc.br)

Horário de atendimento: Segunda-feira das 17:00 às 19:00 por videoconferência (sala virtual a ser definida)

**III. PRÉ-REQUISITO(S)**

CÓDIGO	NOME DA DISCIPLINA
-	-

**IV. CURSO(S) PARA O(S) QUAL(IS) A DISCIPLINA É OFERECIDA**

Bacharelado em Engenharia de Computação

**V. JUSTIFICATIVA**

Um bacharel em Engenharia de Computação que queira trabalhar com programação, certamente usa um compilador (ou um interpretador) como ferramenta para traduzir um programa escrito em uma linguagem de alto nível para um programa escrito em linguagem de máquina. Assim, o estudo da construção de compiladores é fundamental para os engenheiros de computadores, pois o domínio dessa ferramenta leva ao desenvolvimento de programas portáteis de alto desempenho e qualidade.

**VI. EMENTA**

Linguagens-fonte, objeto, de alto-nível e de baixo-nível. Especificação de linguagens de programação. Compilação e interpretação. Processadores de linguagens de programação. Máquinas reais e virtuais. Bootstrapping. Análise sintática. Análise de contexto. Ambientes de execução. Geração de código. Otimização de código independente de máquina. Otimização de código dependente de máquina.

**VII. OBJETIVOS**

**Objetivo Geral**

- Capacitar o aluno na síntese, análise e manipulação de especificações de linguagens de programação de alto nível, assim como no emprego de técnicas de implementação de processadores de linguagens

**Objetivos Específicos**

- Estudar e conhecer os princípios de um compilador;
- Estudar o processo de análise léxica e semântica em um compilador;
- Estudar o processo de geração de código intermediário e código objeto final;
- Estudar o processo de otimização de código intermediário e código objeto final.

## VIII. CONTEÚDO PROGRAMÁTICO

Conteúdo Teórico seguido de Conteúdo Prático com desenvolvimento em computador

### UNIDADE 1: Introdução

- Apresentação da disciplina (ementa, bibliografia, metodologia e avaliações)
- Introdução aos compiladores
- Fases de um compilador

### UNIDADE 2: Análise Léxica

- Expressão regular
- Reconhecedores
- Autômatos finitos para análise léxica

### UNIDADE 3: Análise Sintática

- Gramáticas livres de contexto
- Análise sintática top-down e bottom-up
- Conjuntos FIRST e FOLLOW
- Analisador sintático LR
- Reconhecedores

### UNIDADE 4: Análise Semântica

- Atributos semânticos herdados e sintetizados
- Esquemas S e L atributos

### UNIDADE 5: Geração e Otimização de Código

- Geração de código intermediário
- Otimização de código intermediário
- Geração de código objeto
- Otimização de código objeto

## IX. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA

1. A disciplina será ministrada com aulas expositivas fornecendo os componentes teóricos. Material de apoio postado no Moodle. Desenvolvimento de trabalho e exercícios;
2. Atividades práticas em laboratório visando a construção parcial de um compilador.

### Requisitos de infraestrutura necessários para ministrar as aulas:

- Acesso à Internet;
- Ambiente Virtual de Aprendizagem – Moodle;
- Disponibilidade de um sistema de vídeo conferência.

## X. METODOLOGIA E INSTRUMENTOS DE AVALIAÇÃO

- A verificação do rendimento escolar compreenderá **frequência e aproveitamento** nos estudos, os quais deverão ser atingidos conjuntamente. Será obrigatória a frequência às atividades correspondentes a cada disciplina, no mínimo a 75% das mesmas (Frequência Suficiente - FS), ficando nela reprovado o aluno que não comparecer a mais de 25% das atividades (Frequência Insuficiente - FI).
- A nota mínima para aprovação na disciplina será  $MF \geq 6,0$  (seis) e Frequência Suficiente (FS). (Art. 69 e 72 da Res. nº 17/CUn/1997).
- O aluno com Frequência Suficiente (FS) e média das notas de avaliações do semestre MF entre 3,0 e 5,5 terá direito a uma nova avaliação no final do semestre (REC), exceto as atividades constantes no art.70, § 2º. A Nota Final (NF) será calculada por meio da média aritmética entre a média das notas das avaliações parciais (MF) e a nota obtida na nova avaliação (REC). (Art. 70 e 71 da Res. nº 17/CUn/1997).

$$NF = \frac{(MF + REC)}{2}$$

- Ao aluno que não efetuar às avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero). (Art. 70, § 4º da Res. nº 17/CUn/1997)
- Avaliações

- o Primeira avaliação (P1): Prova escrita referente aos conteúdos da Unidade 1 até a Unidade 2;
- o Segunda avaliação (P2): Prova escrita referente ao conteúdo das Unidades 3, 4 e 5;
- o Trabalho Prático de Compiladores (TP);
- o Os requisitos do trabalho serão divulgados conforme cronograma da disciplina.

A Média Final (MF) será calculada da seguinte forma:

$$MF = P1 \times 0.33 + P2 \times 0.33 + TP \times 0.34$$

- A avaliação no final do semestre (REC) seguirá a mesma regra das avaliações P1 e P2.

- O registro de frequência será efetuado para aulas assíncronas e síncronas. No primeiro caso serão disponibilizadas atividades com tempo de execução de 48 horas em que, a partir da execução destas, os alunos terão a presença registrada. Para o segundo caso ao final das aulas será realizado o registro. Na eventual impossibilidade do aluno estar presente será aplicada a regra da aula assíncrona.

#### Observações:

#### Avaliação de recuperação

- Não há avaliação de recuperação nas disciplinas de **caráter prático** que envolve atividades de laboratório (Res.17/CUn/97).

#### Nova avaliação

- O aluno, que por motivo de força maior e plenamente justificado, deixar de realizar atividades avaliativas previstas no plano de ensino, deverá formalizar pedido à Chefia do Departamento de Ensino ao qual a disciplina pertence, dentro do prazo de 3 (três) dias úteis, apresentando documentação comprobatória. O pedido de nova avaliação deverá ser formalizado na Secretaria Integrada de Departamentos.

### XI. CRONOGRAMA TEÓRICO/PRÁTICO

AULA (semana)	DATA	ASSUNTO
1ª	04/03/2020 a 07/03/2020	Apresentação da disciplina – UNIDADE 1: Apresentação da disciplina (ementa, bibliografia, metodologia e avaliações) Introdução aos compiladores
2ª	09/03/2020 a 14/03/2020	UNIDADE 1: Introdução aos compiladores Fases de um compilador
3ª	31/08/2020 a 05/09/2020	UNIDADE 1: Sincronização e Revisão (aula assíncrona e síncrona)
4ª	07/09/2020 a 12/09/2020	UNIDADE 2: Expressão regular e Reconhecedores (aula assíncrona e síncrona)
5ª	14/09/2020 a 19/09/2020	UNIDADE 2: Autômatos finitos para análise léxica (aula assíncrona e síncrona)
6ª	21/09/2020 a 26/09/2020	UNIDADE 1 e 2: Exercícios (aula assíncrona e síncrona)
7ª	28/09/2020 a 03/10/2020	<b>Revisão (aula síncrona) e PROVA I (Unidade 1 até 2)</b> (atividade assíncrona)
8ª	05/10/2020 a 10/10/2020	UNIDADE 3: Gramáticas livres de contexto (atividade assíncrona)
9ª	12/10/2020 a 17/10/2020	UNIDADE 3: Análise sintática top-down e bottom-up (aula assíncrona e síncrona)
10ª	19/10/2020 a 24/10/2020	UNIDADE 3: Conjuntos FIRST e FOLLOW (aula assíncrona e síncrona)
11ª	26/10/2020 a 31/10/2020	UNIDADE 3: Analisador sintático LR e Reconhecedores (aula assíncrona e síncrona)
12ª	02/11/2020 a 07/11/2020	UNIDADE 4: Atributos semânticos herdados e sintetizados Trabalho Prático (aula assíncrona e síncrona)
13ª	09/11/2020 a 14/11/2020	UNIDADE 4: Esquemas S e L atributos Trabalho Prático (aula assíncrona e síncrona)
14ª	16/11/2020 a 21/11/2020	UNIDADE 3 e 4: Exercícios Trabalho Prático (aula assíncrona e síncrona)

15 <sup>a</sup>	23/11/2020 a 28/11/2020	<b>Revisão (aula síncrona) e PROVA II (Unidades 3 até 4)</b> (atividade assíncrona)
16 <sup>a</sup>	30/11/2020 a 05/12/2020	UNIDADE 5: Geração de código intermediário Otimização de código intermediário Apresentação de trabalhos (atividade síncrona)
17 <sup>a</sup>	07/12/2020 a 12/12/2020	UNIDADE 5: Geração de código objeto Otimização de código objeto Apresentação de trabalhos (atividade síncrona)
18 <sup>a</sup>	14/12/2020 a 19/12/2020	<b>Prova de recuperação e divulgação das notas</b> (atividades assíncrona)

**Obs:** O calendário está sujeito a pequenos ajustes de acordo com as necessidades das atividades desenvolvidas.

## XII. Feriados previstos para o semestre 2020.1:

DATA	
07/09/2020	Independência do Brasil (Segunda-feira)
12/10/2020	Nossa Senhora Aparecida (Segunda-feira)
28/10/2020	Dia do Servidor Público (Lei nº 8.112 – art. 236) (Quarta-feira)
02/11/2020	Finados (Segunda-feira)
15/11/2020	Proclamação da República (Domingo)

## XIII. BIBLIOGRAFIA BÁSICA

1. AHO, Alfred V. et al. Compiladores: Princípios, Técnicas e Ferramentas. 2a ed. São Paulo: Pearson Addison Wesley, 2008.
2. PRICE, Ana Maria de A.; TOSCANI, Simão S. Implementação de Linguagens de Programação: Compiladores. 3a ed. Porto Alegre. Bookman, 2008.
3. ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e Java. 2. ed. São Paulo: Pearson Prentice Hall, 2008.

## XIV. BIBLIOGRAFIA COMPLEMENTAR

SEBESTA, Robert W. Conceitos de linguagens de programação. 9. ed. Porto Alegre: Bookman, 2011.

PATTERSON, David A.; HENNESSY, John L. Organização e projeto de computadores: a interface hardware/software. 3. ed. Rio de Janeiro: Elsevier, 2005.

SANTOS, Rafael. Introdução à programação orientada a objetos usando JAVA. Rio de Janeiro: Campus, 2003.

Os livros acima citados constam na Biblioteca Universitária e Setorial de Araranguá. Algumas bibliografias também podem ser encontradas no acervo da disciplina, via sistema Moodle.

\_\_\_\_\_  
Professor da Disciplina

Aprovado na Reunião do Colegiado do Curso em: \_\_\_\_/\_\_\_\_/\_\_\_\_

\_\_\_\_\_  
Coordenador do Curso