

CURSO:	ENGENHARIAS	PROFESSOR:	Fábio Macêdo Mendes
DISCIPLINA:	Computação Básica	SEMESTRE/ANO:	02/2016
C HORÁRIA:	90 h	CRÉDITOS:	06

PLANO DE ENSINO

1. OBJETIVOS DA DISCIPLINA

Capacitar os alunos para utilizar e compreender os conceitos básicos de programação e compreensão e análise de algoritmos. Habilitar os alunos a formular problemas simples, mas não triviais, em forma de algoritmos e implementá-los na linguagem de programação C.

2. EMENTA DO PROGRAMA

1. História do computador e da resolução de problemas computacionais
2. Variáveis e tipos básicos
3. Estruturas de decisão
4. Laços
5. Funções
6. Estruturas de dados fundamentais
7. Disposição dos objetos na memória
8. Ponteiros
9. Sintaxe do C

3. HORÁRIO DAS AULAS, AVALIAÇÕES E ATENDIMENTO

Aulas teóricas e de exercícios: segundas, quartas e sextas-feiras

Atendimento e monitoria: a definir.

4. METODOLOGIA

O método básico aplicado é o de aulas práticas voltadas à implementação de programas que podem conter partes expositivas orientadas à discussão da teoria. As aulas serão complementadas com atividades de exercícios e demandas extra-classe. Estas atividades serão desenvolvidas com acompanhamento presencial do professor ou através da Plataforma Moodle (<http://www.aprender.unb.br>) e outros sítios externos.

5. CRITÉRIOS DE AVALIAÇÃO

Pontos e estrelas

A avaliação do curso segue uma metáfora de jogos em que a menção final é calculada a partir dos *pontos* e *estrelas* coletados por cada aluno ao longo do curso. O pontos consistem na parte obrigatória da avaliação, e são distribuídos em um total de até 10.000 pontos. O aluno que obter a pontuação completa do curso é aprovado com a menção máxima. A tabela de conversão entre pontuação e menção é a usual: 9.000pts: **SS**, 7.000pts: **MS**, 5.000pts: **MM**, 3.000pts: **MI** e menos que isto **II**.

A distribuição de pontos ao longo do curso é dada pela equação:

$$P_{final} = \frac{N_{obr}}{N_T} (P_1 + P_2 + PE) ,$$

onde P1 e P2 consistem na nota das provas 1 e 2, valendo 4.000 pontos cada, e PE consiste na pontuação total devido a exercícios ministrados em sala de aula. O fator multiplicativo consiste no número de exercícios obrigatórios resolvidos corretamente pelo aluno dividido pelo número total destes exercícios. O aluno pode tentar resolver estes exercícios quantas vezes for necessário, sendo que o sistema de avaliação automática fornece dicas sobre os eventuais erros cometidos na resolução destes exercícios.

As estrelas são coletadas em atividades optativas e podem ser convertidas em “*poderes especiais*” ou em pontos ao final do curso. Os poderes conferidos por estrelas ajudam o aluno melhorar a nota e serão descritos com mais cuidado no Moodle da disciplina (ex: abonar uma falta, direito a fazer a prova substitutiva, etc). No final do curso, o aluno pode trocar as estrelas que estão sobrando por nota usando a seguinte regra:

- Cada estrela equivale 100 pontos.
- Soma-se a pontuação de todas as estrelas PE e calcula-se a nova pontuação final de acordo com

$$P'_{final} = PE + \frac{10.000 - PE}{10.000} P_{final}$$

As estrelas sempre aumentam a nota final e garantem uma pontuação mínima PE independente da nota obtida nas provas e trabalhos.

6. PROVA SUBSTITUTIVA E FALTAS

Cada aluno possui uma “barra de vidas” com 13 vidas. Cada falta implica na perda de uma vida. Uma vida pode ser recuperada gastando-se 5 estrelas. Faltas com justificativa médica **não** serão abonadas. A prova substitutiva será aplicada apenas em caso de falta justificada no dia da prova. O aluno deve apresentar a justificativa na aula seguinte à prova ou quando terminar a licença médica. O aluno que atingir **zero** vidas estará automaticamente com menção **SR**.

7. CÓDIGO DE ÉTICA E CONDUTA

As avaliações serão realizadas com auxílio do computador no laboratório de informática. Todas as submissões serão processadas por um programa de detecção de plágio. Qualquer atividade onde for detectada a presença de plágio será anulada sem a possibilidade de substituição. Não será feita qualquer distinção entre o aluno que forneceu a resposta para cópia e o aluno que obteve a mesma.

A análise de plágio será feita inicialmente apenas nas provas. O aluno que se envolver em plágio em uma das provas poderá ter todas as outras atividades analisadas.

8. CRONOGRAMA DE ATIVIDADES

Semana	Dia	Aula
1	8/8/2016	Início das aulas – Apresentação do curso <ul style="list-style-type: none"> • Moodle • Pytuguês • Ambiente Linux • Codeschool
	10/8/2016	Ambiente de programação <ul style="list-style-type: none"> • Pytuguês, comandos básicos e terminal iterativo • Pytuguês como uma calculadora • Programa de computador como sequência de ações • Desenhando figuras geométricas

Semana	Dia	Aula
	12/8/2016	Variáveis inteiras e representação binária <ul style="list-style-type: none"> Representação binária de números inteiros Conversão para outros sistemas numéricos
2	15/8/2016	Repetições <ul style="list-style-type: none"> Repetições simples Repetições em sequências Repetições condicionais Acumuladores
	17/8/2016	Execução condicional <ul style="list-style-type: none"> Execução condicional Execução condicional dentro de laços Condições complexas
	19/8/2016	Algoritmos <ul style="list-style-type: none"> Algoritmos como sequência de passos Representação por fluxogramas Convertendo operações comuns em algoritmos
3	22/8/2016	Entrada e saída de dados <ul style="list-style-type: none"> Variáveis tipo string Tipos de variáveis Entrada e saída de dados
	24/8/2016	Funções e sub-rotinas <ul style="list-style-type: none"> Chamando funções Definindo funções e sub-rotinas Reutilização de código Documentação de funções
	26/8/2016	Sintaxe básica do Python <ul style="list-style-type: none"> Laços, condicionais, funções Entrada e saída Executando no terminal Turtle art
4	29/8/2016	Listas <ul style="list-style-type: none"> Extração e alteração de elementos Adição e remoção de elementos Pesquisa e ordenamento Iteração
	31/8/2016	Dicionários <ul style="list-style-type: none"> Mapeamento de elementos Iteração Operações com dicionários
	2/9/2016	Linguagem C <ul style="list-style-type: none"> Linguagens de programação Tipagem estática e declaração de variáveis Sintaxe do C Compilação de programas
5	5/9/2016	Sintaxe básica do C <ul style="list-style-type: none"> Função main() Declaração de variáveis Utilizando o TCC Compilando no GCC
	7/9/2016	<i>Feriado – Independência do Brasil</i>
	9/9/2016	Laboratório de Jogos

Semana	Dia	Aula
		<ul style="list-style-type: none"> Pygame Zero Sprites Eventos Loop principal
6	12/9/2016	Entrada e saída de dados <ul style="list-style-type: none"> `printf()` e `scanf()` função `getchar()`, `putchar()` e similares entradas incorretas
	14/9/2016	Comparação de programas: Python vs. Pytuguês vs. C <ul style="list-style-type: none"> Entrada e saída Condicionais Funções
	16/9/2016	Tipos numéricos <ul style="list-style-type: none"> Tipos inteiros Python vs. C Notação científica binária Ponto flutuante
7	19/9/2016	Funções recursivas <ul style="list-style-type: none"> Recursividade Exercício: Fatorial recursivo Exercício: Fibonacci recursivo
	21/9/2016	Funções recursivas <ul style="list-style-type: none"> Substituindo laços Escopo de variáveis em funções Funções de funções
	23/9/2016	Análise de complexidade <ul style="list-style-type: none"> Complexidade de algoritmos Notação Big-O Memória vs. CPU
8	26/9/2016	Repetições – “while” <ul style="list-style-type: none"> Sintaxe básica Variáveis contadoras Variáveis acumuladoras
	28/9/2016	Repetições – “for” <ul style="list-style-type: none"> Sintaxe básica Convertendo o “for” em “while” Quebra de laços
	30/9/2016	Compilação <ul style="list-style-type: none"> Header files Etapas de compilação em C Makefiles
9	3/10/2016	Arrays <ul style="list-style-type: none"> Alocação estática de memória Criação de arrays Acesso a elementos Overflow Diferenças entre arrays e listas
	5/10/2016	Strings <ul style="list-style-type: none"> Lista de caracteres Comparação Caractere nulo e comprimento de uma string

Semana	Dia	Aula
	7/10/2016	Codificação de caracteres <ul style="list-style-type: none"> Variáveis do tipo char Representação binária Padrão ASCII Unicode e codificação UTF-8
10	10/10/2016	Strings <ul style="list-style-type: none"> Operações básicas com strings Principais funções em “string.h”
	12/10/2016	<i>Feriado – Nossa senhora aparecida</i>
	14/10/2016	<i>Feriado – Dia do servidor público</i>
11	17/10/2016	Trabalhando com arquivos <ul style="list-style-type: none"> Função “open()” e modos de abertura de um arquivo Processamento de dados em arquivos longos
	19/10/2016	Trabalhando com arquivos <ul style="list-style-type: none"> Inicialização de um arquivo “fscanf()” e “fprintf()” Arquivos especiais stdin e stdout
	21/10/2016	Revisão para a prova
12	24/10/2016	Prova I – Conceitos básicos de programação (até semana 10)
	26/10/2016	Tipos definidos pelo usuário <ul style="list-style-type: none"> Novos tipos e “typedef” Estruturas em C Comando “class” em Python
	28/10/2016	Arquitetura básica de computadores <ul style="list-style-type: none"> CPU Tipos de memória Endereçamento
13	31/10/2016	Ponteiros <ul style="list-style-type: none"> Disposição de objetos na memória Operadores de referenciamento e de deferenciamento Passagem de argumentos por valor ou referência
	2/11/2016	<i>Feriado - Finados</i>
	4/11/2016	“Stack” e “Heap” <ul style="list-style-type: none"> Funcionamento de uma chamada de função Variáveis da “stack” Variáveis da “heap” “Stack overflow”
14	7/11/2016	Ponteiros <ul style="list-style-type: none"> Ponteiros e “arrays” Álgebra de ponteiros
	9/11/2016	Alocação dinâmica <ul style="list-style-type: none"> Funções “malloc()” e “free()” Gerenciamento de memória
	11/11/2016	Tratamento de erros <ul style="list-style-type: none"> Códigos de erro em C Exceções em Python Passagem de parâmetros por referência Problemas e “perigos” dos ponteiros
15	14/11/2016	Comparação de performance <ul style="list-style-type: none"> Processamento numérico

Semana	Dia	Aula
		<ul style="list-style-type: none"> Criando benchmarks Chamando C a partir do Python
	16/11/2016	Laboratório científico <ul style="list-style-type: none"> Matplotlib Numpy
	18/11/2016	Simulações numéricas <ul style="list-style-type: none"> Simulando um sistema físico
16	21/11/2016	Aula de revisão
	23/11/2016	Prova II – programação avançada (até semana 14)
	25/11/2016	
17	28/11/2016	Laboratório Web <ul style="list-style-type: none"> Introdução ao Django Tecnologias Web: HTML, CSS e JavaScript Views
	30/11/2016	Laboratório Web (continuação) <ul style="list-style-type: none"> Classes e banco de dados Templates Django Admin
	2/12/2016	
18	5/12/2016	Prova substitutiva
	7/12/2016	
	9/12/2016	Fim das aulas

Obs.: O cronograma está sujeito a alterações.

9. BIBLIOGRAFIA

BÁSICA:

DAMAS, Luis M. D. Linguagem C, 10. ed, Rio de Janeiro, LTC, 2007.

MENEZES, Nillo N. C. *Introdução à programação com Python*, Novatec, 2010

VINE, Michael. C Programming for the Absolute Beginner, 2. ed, Ed. Boston MA, USA Course Technology, 2007. Disponível na eBrary

COMPLEMENTAR:

MILLER, Brad e **RANUM**, David. *How to Think Like a Computer Scientist, Learning with Python: Interactive Edition*, <http://interactivepython.org/courselib/static/thinkcspy/index.html>
 Learn C The Hard Way, <http://c.learnthecodethehardway.org/book/>