



.Brothers.E01.1080p.5.1Ch.BluRay.ReEnc-DeeJayAhmedUNIVERSID
FEDERAL DE OURO PRETO
PRÓ-REITORIA DE GRADUAÇÃO
PLANO DE ENSINO



Nome do Componente Curricular em português: Programação de Computadores I	Código: CSI101			
Nome do Componente Curricular em inglês: Computer Programming I	Turma: SI-01			
Nome e sigla do departamento: Departamento de Computação e Sistemas (DESCI)	Unidade acadêmica: ICEA			
Nome do docente: Darlan Nunes de Brito				
<table border="1"><tr><td>Carga horária semestral 72 horas/aula</td><td>Carga horária semanal teórica 02 horas/aula</td><td>Carga horária semanal prática 02 horas/aula</td></tr></table>		Carga horária semestral 72 horas/aula	Carga horária semanal teórica 02 horas/aula	Carga horária semanal prática 02 horas/aula
Carga horária semestral 72 horas/aula	Carga horária semanal teórica 02 horas/aula	Carga horária semanal prática 02 horas/aula		
Data de aprovação na assembleia departamental: 15/12/2020				
Ementa: Arquitetura e funcionamento básico de um computador. Conceito de Algoritmo. Estratégias básicas de solução de problemas por meio de algoritmos. Conceito de programa. Conceitos básicos de linguagens de programação de alto nível: tipos primitivos, variáveis, atribuição, operadores, expressões. Sequenciamento de instruções. Controle de fluxo de execução: estruturas de seleção e repetição. Vetores. Ponteiros. Procedimentos e funções. Alocação de memória. Tipos estruturados de dados. Entrada e saída de dados. Implementação de programas de pequeno porte.				
Conteúdo programático: <ol style="list-style-type: none">Arquitetura e funcionamento básico de um computador.<ol style="list-style-type: none">Modelo de Von-NeumamConceito de algoritmo.<ol style="list-style-type: none">Representação de algoritmos: Narração descritiva, pseudo-código e fluxograma;Conceito de programa.<ol style="list-style-type: none">Lógica de programação;Regras de sintaxe;Programa fonte; Compilador; Programa objeto; Programa executável.Tipos de dados primitivos, conceitos de variáveis e constantes, inicialização de variáveis, operadores e expressões, coerção de tipos.Entrada e saída de dados.Controle do fluxo de execução.<ol style="list-style-type: none">Bloco de comandos;Desvio condicional simples: if-then e if-then-else;				

- 3.3. Desvio condicional múltiplo: case ou switch;
- 3.4. Laços: com teste no começo (while), com contador de passo (for) e com teste no final (repeat).
4. Funções.
 - 4.1. Definição de funções;
 - 4.2. Parâmetros e retorno de funções;
 - 4.3. Passagem de parâmetro: por referência e por valor.
 - 4.4. Escopo de variáveis: escopo local e escopo global.
 - 4.5. Gerenciamento de memória: Pilha de chama de funções.
 - 4.6. Recursão.
5. Vetores
 - 5.1. Definição
 - 5.2. Alocação
 - 5.3. Strings: Representação, manipulação: funções de processamento de strings.
 - 5.4. Vetores multidimensionais.
6. Alocação dinâmica de memória:
 - 6.1. Alocação de variáveis e vetores;
 - 6.2. Heap de memória;
 - 6.3. Liberação da memória;
7. Algoritmos de busca e ordenação de dados em vetores: bubbleSort, busca sequencial, busca binária.
8. Tipos de dados definidos pelo usuário

Objetivos:

Objetivos Gerais

Introduzir os conceitos associados à informática e a sua utilização em um contexto de suporte às atividades do aluno e do futuro profissional. Ajudar a desenvolver o raciocínio lógico e a capacidade de abstração do aluno através da solução de problemas técnicos e científicos, utilizando eficientemente um sistema de computação.

Objetivos Específicos

Ao final do curso, espera-se que o aluno seja capaz de:

- Especificar um pequeno sistema de computação capaz de ajudar na solução de um problema a partir da caracterização da situação e do fornecimento de descrições de componentes;
- Resolver problemas contextualizados, utilizando a linguagem de programação C;
- Decompor a solução de um problema em módulos capazes de serem implementados em um computador, adotando uma disciplina de programação, reconhecendo as limitações e dificuldades do processo de programação;

Desenvolver algoritmos estruturados para a solução de problemas;

Metodologia:

A disciplina será desenvolvida na modalidade Ensino Remoto (ER), por meio de tecnologias digitais de informação e comunicação com o docente e os discentes. Interações assíncronas e síncronas serão realizadas para o desenvolvimento das atividades previstas.

As principais ações podem ser descritas como segue:

- Preparação de videoaulas, texto, notas de aula e outros conteúdos complementares.

- Indicação de sites, vídeos, podcasts e demais conteúdos para complementar o aprendizado, além de contribuir para as discussões.
- Discussões e interações via videoconferência e no ambiente virtual de aprendizagem (Moodle) serão promovidas sempre que necessário.

Recursos:

Para o desenvolvimento da aprendizagem serão adotados, concomitantemente e em todos os tópicos da disciplina, os seguintes recursos de apoio didático-pedagógico:

- Ferramentas do ambiente virtual de aprendizagem
- Videoaulas
- Fóruns de discussão no Moodle
- e-mail da UFOP
- Códigos e afins no GitHub
- Vídeos online produzidos por terceiros
- Podcasts e similares
- Caso seja necessário, videoconferência (via Google Meet) com apresentação multimídia (slides) e/ou desenvolvimento de código com aulas expositivas dialogadas
- Dentre outros

Para os discentes acompanharem as aulas e realizarem as provas são necessários os seguintes recursos: computador/notebook com as aplicações instaladas, webcam e microfone.

Controle de frequência:

O acompanhamento da frequência das aulas assíncronas será realizado por meio do acesso dos discentes ao Moodle e da entrega das atividades. Para as atividades síncronas, a frequência será apurada no decorrer da aula.

Atividades avaliativas:

Descrição da avaliação	Peso da avaliação (%)	Data	Conteúdo avaliado
Prova 1	15%	30/06	Conteúdo das aulas de 01 a 09
Prova 2	15%	28/07	Conteúdo das aulas de 01 a 20
Prova 3	15%	25/08	Conteúdo das aulas de 01 a 28
Questionários	55%	N/A	Conforme o conteúdo em questão.
Exame Especial Parcial	Conforme a prova em questão	03/09	Conforme o conteúdo em questão.
Exame Especial Total	100%	03/09	Toda a matéria.

Observações:

- As listas de exercícios contemplarão diversos conteúdos e deverão ser entregues no decorrer do semestre.
- Todas as atividades são individuais.
- Os procedimentos e o cronograma das avaliações poderão ser alterados a partir de discussões coletivas com os discentes.

- As provas serão realizadas de maneira síncrona e terão questões teóricas e práticas. Para garantir a segurança dos discentes, as câmeras devem estar ligadas durante a realização da prova.

Horário de Aula

Dia da semana	Horário	Tecnologia	Interação
Quarta-feira	13:30 às 15:10 hs	Moodle UFOP	Síncrona/Assíncrona
Sexta-feira	13:30 às 15:10 hs	Moodle UFOP	Síncrona/Assíncrona

Horário de Atendimento

Dia	Horário	Tecnologia	Interação
Terça-feira	18:00 às 18:50 hs	Google Meet	Síncrona
Quinta-feira	18:00 às 18:50 hs	Google Meet	Síncrona

Obs.: Nas terças e quintas-feiras o professor estará disponível on-line via Google Meet para tirar dúvidas dos estudantes. Os alunos poderão enviar perguntas por meio do Fórum do Moodle (forma de texto), pela ferramenta mensagem do Moodle ou via email. O retorno será enviado pelo professor em até 1 dia útil. Todos os atendimentos serão gravados em vídeo e disponibilizados no Moodle.

Contato

E-mail do Professor	darlan@ufop.edu.br
---------------------	--------------------

Planejamento das Aulas

(sujeito a mudanças no decorrer do semestre)

A modalidade da aula é definida pelo Tipo, sendo S – Síncrona e A – Assíncrona.

Aula	Prática/ Teórica	Data	Tipo	Conteúdo Previsto
1	T	26/5	S	Aula introdutória da disciplina (ementa, conteúdo programático, avaliações etc.).
2	P/T	28/5	S	Introdução, conceitos básicos, sistema de numeração, arquitetura, processadores, periféricos, sistemas operacionais. Sistemas aplicativos.
3	T	2/6	A	Estrutura léxica, sintática e semântica. Linguagem C, compilador, IDE (CodeBlocks). Primeiro programa em C.
4	P	4/6	A	Declaração de variáveis, tipos básicos, estruturas de controle básicas. Entrada e saída básica.
5	T	9/6	A	Regras básicas para a construção de algoritmos legíveis (endentação, nomes de variáveis etc.).
6	P	11/6	S	Exercícios sobre entrada e saída.
7	T	16/6	A	Estruturas condicionais: decisão com expressões lógicas e alternativas.
8	P	18/6	S	Exercícios com expressões lógicas
9	P/T	23/6	A	Comandos de Repetição

10	T	25/6	S	Exercícios de Comandos de Repetição
11	P	30/6	S	Prova 1
12	T	2/7	A	Sub-rotinas (Procedimentos e funções).
13	P	7/7	A	Exercícios básicos com procedimentos e funções.
14	P	9/7	A	Vetores e Strings.
15	T	14/7	S	Exercícios de Vetores
16	P	16/7	S	Exercícios de Strings.
17	P/T	21/7	A	Laços encaixados e Matrizes.
18	T	23/7	S	Exercícios de Laços encaixados e Matrizes.
19	T	28/7	S	Prova 2
20	T	30/7	A	Recursão
21	T	4/8	A	Ponteiros
22	T	6/8	A	Introdução à alocação dinâmica de memória.
23	P	11/8	A	Exercícios de ponteiros e alocação dinâmica de memória.
24	T	13/8	A	Introdução às estruturas heterogêneas (<i>structs</i>) e arranjos de estruturas.
25	T	18/08	S	Exercícios de estruturas
26	P/T	20/8	A	Algoritmos de busca e ordenação de dados em vetores
27	P/T	25/8	S	Prova 3
27	T	27/8	A	Revisão da prova
28	P/T	03/9	S	Exames Especiais – Total e Parcial

Bibliografia básica:

1. DEITEL P. J., DEITEL H. M., C: como programar, 6ª edição, São Paulo: Pearson, 2011. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/2660>.
2. DEITEL P. J., DEITEL H. M., C++: como programar, 6ª edição, São Paulo: Pearson, 2006. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/338>.
3. MIZRAHI V.V. Treinamento em Linguagem C, 2ª edição, Prentice-Hall, 2008. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/2781>

Bibliografia complementar:

1. LUIZ, A.; FORBELLONE V.;EBERSPACHER H. F. Lógica de programação: a construção de algoritmos e estruturas de dados, 3ª edição, Pearson, 2005. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/323>
2. DEITEL P. J., DEITEL H. M., Java: como programar, 8ª edição, São Paulo: Prentice Hall, 2010. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/325>
3. Algoritmos – UFJF. Material - Turma X. Disponível em: <https://sites.google.com/site/algoritmosufjf/ere/turma-x/material-x-ere>. Acessado em 08 dez 2020.
4. Algoritmos da UFJF. Canal da disciplina de Algoritmos da UFJF. Disponível em: <https://www.youtube.com/channel/UCCnuaoGvAmw-xRuaLN8CmKg/playlists>. Acessado em 08 dez 2020.
- MALAN, David J.; LLOYD, Doug; YU, Brian. CS50's Introduction to Computer Science. Disponível em: <https://www.edx.org/course/cs50s-introduction-to-computer-science>. Acessado em 08 dez 2020.2010.
5. SENAC. Construção de Algoritmos. São Paulo: SENAC, 2002.

