

Escola/ Câmpus:	Politécnica - Curitiba			
Curso:	Ciência da Computação e Cibersegurança	Ano/Semestre:	2022/2	
Código/Nome da disciplina:	PCP113A - Programação Imperativa			
Carga Horária:	80 horas			
Requisitos:	Raciocínio Algoritmo			
Créditos:	4	Período: 2	Turma: U	Turno: Diurno e Noturno
Professor Responsável:	Alcides Calsavara e Edson Justino			

1. Ementa

A disciplina, de natureza teórico-prática, é ofertada no segundo período dos cursos de Ciência da Computação e Cibersegurança. O estudante aplica os conceitos de programação imperativa na resolução de problemas computacionais estruturados. Ao término da disciplina, o estudante é capaz de utilizar uma linguagem imperativa de nível médio de abstração, tal como a linguagem C, de forma a considerar as funcionalidades e recursos específicos da linguagem para fins de eficiência dos programas desenvolvidos com relação ao uso de memória e de processador. Para frequentar a disciplina, é necessário ter conhecimento de lógica de programação e de construção de algoritmos.

2. Relação com disciplinas precedentes e posteriores

Esta disciplina é parte integrante do eixo de formação em programação dos cursos. Ela tem a disciplina de Raciocínio Algoritmo como pré-requisito direcional. A disciplina fornece a base conceitual e tecnológica para as seguintes disciplinas posteriores:

1. Conectividade em Sistemas Ciberfísicos
2. Experiência Criativa: criando soluções computacionais
3. Segurança da Informação
4. Performance em Sistemas Ciberfísicos
5. Redes Convergentes
6. Sistemas Operacionais Ciberfísicos
7. Programação Distribuída
8. Construção de Interpretadores

3. Temas de estudo

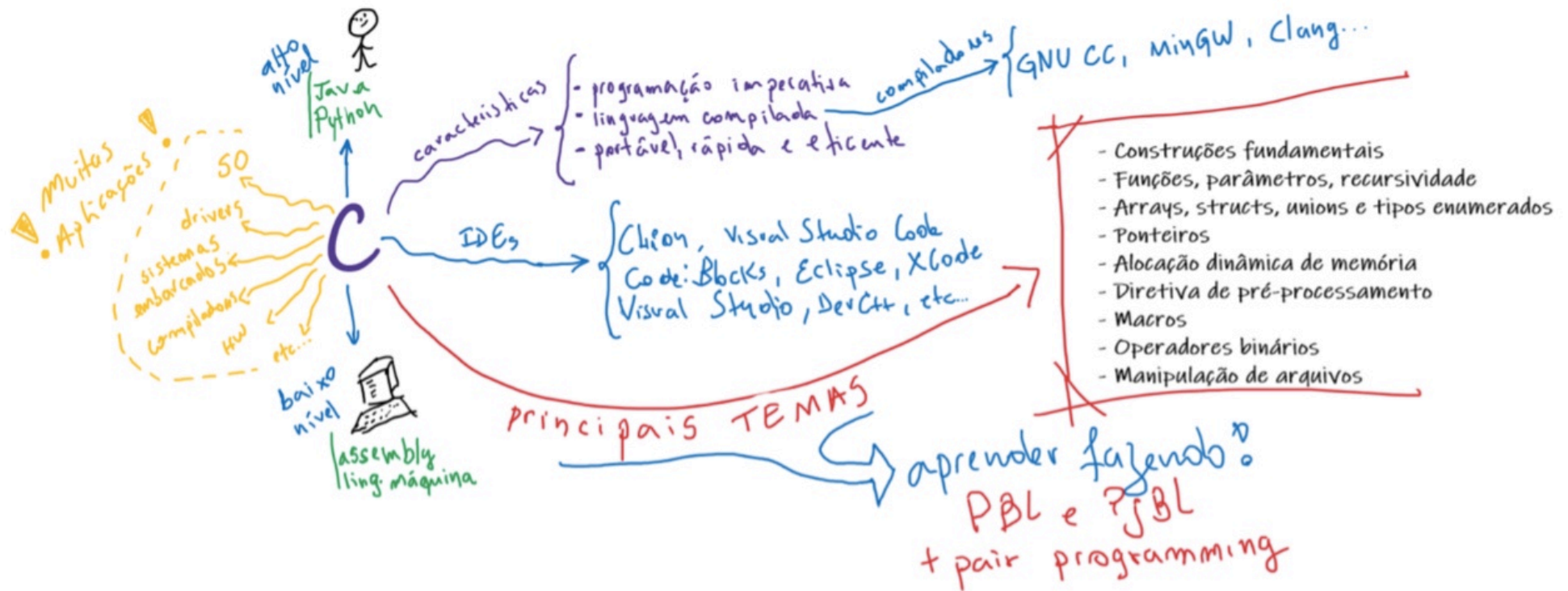
1. Representação abstrata de dados (tipos de dados)
2. Recursos para escrita e leitura de dados
3. Comandos de controle de fluxo
4. Estruturação de programas em funções
5. Boas práticas de codificação e documentação
6. Representação física de dados (memória e endereçamento)
7. Operações sobre bits
8. Arquitetura do sistema de execução
9. Alocação dinâmica de memória
10. Arquivos

4. Resultados de Aprendizagem

Resultados de Aprendizagem	Temas de Estudo	Elemento de Competência (Internaliza, Mobiliza, Certifica) e Competência
RA 1. Desenvolver programas auto-documentados em uma linguagem imperativa fortemente tipada	1 a 5	Competência A: Desenvolver software eficiente e seguro, seguindo padrões e boas práticas, de forma criativa e autônoma. Elementos de Competência: A1. Avaliar o contexto do problema e recursos de hardware e software A2. Projetar soluções de software A3. Implementar software eficiente e seguro de forma criativa e autônoma A4. Avaliar a implementação do software
RA 2. Utilizar a memória e o processador do computador de forma adequada e eficiente por meio de uma linguagem imperativa	6 a 10	Competência A: Desenvolver software eficiente e seguro, seguindo padrões e boas práticas, de forma criativa e autônoma.

		<p>Elementos de Competência:</p> <p>A1. Avaliar o contexto do problema e recursos de hardware e software</p> <p>A2. Projetar soluções de software</p> <p>A3. Implementar software eficiente e seguro de forma criativa e autônoma</p> <p>A4. Avaliar a implementação do software</p>
--	--	---

5. Mapa Mental



(Elaborado por Prof. Henri F. Eberspacher)

6. Metodologia e Avaliação

Alinhamento Construtivo			
Resultado de aprendizagem	Indicadores de desempenho	Processos de Avaliação	Métodos ou técnicas empregados**
RA 1. Desenvolver programas auto-documentados em uma linguagem imperativa fortemente tipada	<p>ID 1.1. Define variáveis e constantes do tipo apropriado</p> <p>ID 1.2. Define tipos de dados enumerados e compostos</p> <p>ID 1.3. Codifica programas em linguagem imperativa fortemente tipada em conformidade com as boas práticas de codificação e de documentação</p>	<p>Avaliação formativa:</p> <p>Resolução de exercícios em sala de aula sobre os conceitos e as técnicas de programação</p> <p><i>Feedback:</i> Coletivo</p> <p>Avaliação somativa:</p> <p>1. Uma série de, aproximadamente, 4 trabalhos semanais realizados em equipes para a aplicação dos conceitos e das técnicas de programação na resolução de problemas (PBL).</p> <p><i>Feedback:</i> Por equipe</p> <p>2. Um projeto realizado individualmente de longa duração (aproximadamente, um mês) para a aplicação dos conceitos e das técnicas de programação para desenvolvimento de um sistema (PjBL).</p> <p><i>Feedback:</i> Individual</p>	<p>1. Aula expositiva dialogada sobre conceitos e técnicas de programação</p> <p>2. Resolução em equipes (TBL) de exercícios em sala de aula sobre os conceitos e as técnicas de programação</p> <p>3. Aplicação dos conceitos e das técnicas de programação na resolução de problemas (PBL - em equipes) e no desenvolvimento de um projeto (PjBL - individual)</p> <p>4. Atividade extra-classe para estudo complementar de temas específicos, incluindo operações sobre arquivos textos</p>

<p>RA 2. Utilizar a memória e o processador do computador de forma adequada e eficiente por meio de uma linguagem imperativa</p>	<p>ID 2.1. Escreve expressões baseadas em operadores lógicos sobre bits, complemento binário e de deslocamento de bits</p> <p>ID 2.2. Define e manipula corretamente memória alocada dinamicamente</p> <p>ID 2.3. Emprega corretamente o modo de passagem por referência nos parâmetros e no retorno de uma função</p>	<p>Avaliação formativa:</p> <p>Resolução de exercícios em sala de aula sobre os conceitos e as técnicas de programação</p> <p><i>Feedback:</i> Coletivo</p> <p>Avaliação somativa:</p> <p>1. Uma série de, aproximadamente, 4 trabalhos semanais realizados em equipes para a aplicação dos conceitos e das técnicas de programação na resolução de problemas (PBL).</p> <p><i>Feedback:</i> Por equipe</p> <p>2. Um projeto realizado individualmente de longa duração (aproximadamente, um mês) para a aplicação dos conceitos e das técnicas de programação para desenvolvimento de um sistema (PjBL).</p> <p><i>Feedback:</i> Individual</p>	<p>1. Aula expositiva dialogada sobre conceitos e técnicas de programação</p> <p>2. Resolução em equipes (TBL) de exercícios em sala de aula sobre os conceitos e as técnicas de programação</p> <p>3. Aplicação dos conceitos e das técnicas de programação na resolução de problemas (PBL - em equipes) e no desenvolvimento de um projeto (PjBL - individual)</p> <p>4. Atividade extra-classe para estudo complementar de temas específicos, incluindo operações sobre arquivos binários</p>
--	--	--	--

Avaliação Somativa:

Tipo de trabalho	Descrição	Forma de realização	Peso na nota do semestre
PBL	Trabalhos semanais (aproximadamente, 10 entregas, todas com o mesmo peso na nota)	Equipes de, no máximo, 4 estudantes	60%
PjBL	Trabalho de longa duração (aproximadamente, um mês) e de tema livre	Individual	40%

Critério de aprovação: nota do semestre igual ou superior a **7** (sete) e frequência mínima de **75%**.

Recuperação:

O estudante pode recuperar a nota de todo e qualquer trabalho:

1. Se um PBL não for entregue no prazo, o estudante poderá entregar com atraso de até duas semanas após o prazo, sendo a nota máxima igual a 7 (sete).
2. Se um PBL for entregue no prazo e a nota atribuída seja inferior a 7 (sete), o estudante poderá refazer a entrega até uma semana após a publicação da nota.
3. Se o PjBL não for entregue no prazo ou caso tenha sido entregue e a nota atribuída seja inferior a 7 (sete), o estudante poderá entregar até a “Semana Estendida de Recuperação”, conforme Calendário Acadêmico da PUCPR, sendo a nota máxima igual a 7 (sete).

7. Cronograma de atividades

Período (horas aula totais, dia, semana, quinzena, mês)	RAs	Atividades pedagógicas	Em aula / TDE	Carga horária da atividade
Semana 1	1	Introdução à linguagem C. Estudo das funções para entrada e saída de dados.	Em aula	4
Semana 2	1	Estudo sobre variáveis de tipos primitivos e operadores.	Em aula	4
Semana 3	1	Estudo dos comandos de desvio de fluxo (if) e de repetição (while e do)	Em aula	4
Semana 4	1	Estudo sobre tipos de dados compostos e o comando switch	Em aula	4
	1	Estudo sobre boas práticas de codificação e documentação	TDE	4
Semana 5	1	Estudo sobre alocação estática de vetores e matrizes e sobre o comando de repetição for	Em aula	4
Semana 6	1	Estudo sobre definição e uso de funções, parâmetros e variáveis globais	Em aula	4
Semana 7	1	Estudo sobre a estruturação de código fonte em arquivos cabeçalho e arquivos de implementação	Em aula	4
	2	Estudo sobre funções para leitura e escrita de dados em arquivo texto	TDE	4 a 6
Semana 8	2	Estudo sobre operadores lógicos sobre bits, complemento binário e deslocamento de bits	Em aula	4
Semana 9	2	Estudo sobre a representação física de dados em memória, endereçamento de memória, uso de ponteiros e operador sizeof	Em aula	4
Semana 10	2	Estudo sobre alocação dinâmica de memória: funções malloc e free	Em aula	4
Semana 11	2	Estudo sobre o modo de passagem por referência para os parâmetros e o retorno de uma função	Em aula	4
	2	Estudo sobre arquivos binários	TDE	4 a 6
Semana 12	1 e 2	Desenvolvimento do PjBL	Em aula	16
Semana 13				
Semana 14				
Semana 15				

8. Bibliografia

Básica:

1. BACKES, André. **Linguagem C Completa e Descomplicada**. Elsevier Editora Ltda. 2013.
2. PINHEIRO, Francisco C. **Elementos de programação em C**. Bookman, 2012.
3. SCHILDT, Herbert. **C Completo e Total**. Berkeley. Pearson, 1997.

Complementar:

1. STROUSTRUP, B. **A linguagem de programação C++**. Bookman, 2000.
2. DAMAS, L. **Linguagem C**. São Paulo: LTC, 2007.
3. KERNIGHAN, BW; RITCHIE, DM. **C: a linguagem de programação padrão ANSI**. Campus, 1990.
4. MIZRAHI, VV. **Treinamento em linguagem C**. Pearson/Prentice Hall, 2008.
5. PEREIRA, SL. **Algoritmos e lógica de programação em C: uma abordagem didática**. Erica, 2010.

9. Acessibilidade**

Não houve necessidade de adaptação.

10. Adaptações para práticas profissionais**

Não se aplica.

** conforme nota técnica conjunta número 17/2020 CGLNRS/DPR/SERES/SERES