



INSTRUÇÃO PRÁTICA		PI-P014
MÓDULO	PI - PROGRAMAÇÃO IMPERATIVA	
OBJETIVO DA ATIVIDADE		TEMPO
Aplicar os conceitos de programação estruturada, modularização, reutilização de código e organização de programas, utilizando structs e funções em C++. Além disso, a atividade incentiva a prática de resolução de problemas, o desenvolvimento da lógica de programação e a colaboração em grupo, preparando-os para uma transição mais suave para a programação orientada a objetos.		2h
DESCRIÇÃO		
<p>Exercício 1: Crie um repositório para esta atividade.</p> <ul style="list-style-type: none">• Sua equipe deve criar um repositório no GitHub para esta atividade. O repositório deve conter:<ul style="list-style-type: none">◦ Um README descrevendo a atividade e as informações dos membros da equipe◦ Os arquivos de código fonte de cada uma das atividades.• Para realizar a atividade cada um dos membros da equipe deve criar um branch no repositório, onde fará as implementações da parte individual e outro Branch para mesclar os códigos relativos à parte em grupo.• Publicar no Moodle o endereço do repositório e o branch específico em que está desenvolvendo suas atividades• Utilize o fórum do Moodle para tirar suas dúvidas sobre os exercícios. <p>Parte Individual:</p> <p>Exercício 1: Reutilização de Funções</p> <ul style="list-style-type: none">• Escreva uma função em C++ que calcule o fatorial de um número inteiro.• Crie um programa que utilize essa função para calcular e exibir o fatorial de três números diferentes. <p>Exercício 2: Modularização</p> <ul style="list-style-type: none">• Implemente um programa que converta uma temperatura de Celsius para Fahrenheit.• Separe a lógica de conversão em uma função e a entrada/saída em outra função.• Teste a função de conversão com diferentes valores. <p>Exercício 3: Estruturas de Dados com Structs</p> <ul style="list-style-type: none">• Crie uma struct chamada "Ponto" que represente as coordenadas (x, y) de um ponto no plano.• Implemente uma função que calcule a distância entre dois pontos.• Utilize a struct e a função para calcular e exibir a distância entre dois pontos dados.		



Parte em Grupos (até 5 pessoas por grupo):

Exercício 4: Organização de Código

- Em grupo, escolham uma funcionalidade simples para um programa (por exemplo, uma calculadora de operações matemáticas).
- Dividam o programa em diferentes funções, atribuindo a cada membro do grupo a responsabilidade por uma função específica.
- Juntem as funções criadas por cada membro e criem um programa completo que utilize todas elas.
- Usem os recursos do Git para dividir e juntar os códigos.

Exercício 5: Abstração e Modularização

- Em grupo, escolham um objeto do mundo real (por exemplo, um carro) e identifiquem suas características e comportamentos.
- Dividam o objeto em partes menores e atribuam a cada membro a responsabilidade por modelar uma parte como uma struct.
- Juntem as structs criadas por cada membro para formar uma representação completa do objeto.

REFERÊNCIAS

Luiz Paulo Moreira Silva. O que é plano cartesiano? Brasil Escola Online. Disponível em: <https://brasilecola.uol.com.br/o-que-e/matematica/o-que-e-plano-cartesiano.htm> Acessado em 04 de agosto de 2023.

FARRER, Harry et al. **Algoritmos Estruturados**. 3 ed. Rio de Janeiro: LTC, 1999. 260 p.

SCHILDT, Herbert. **C Completo e Total**. São Paulo: Makron, McGraw-Hill, 1990. 889 p

GUIMARÃES, A. de M; LAGES, N. A. de C. **Algoritmos e Estruturas de Dados**. L.T.C, 1994. 216 p.