

Sumário da aula 8

- Aplicação de uma ficha de avaliação para consolidar os conteúdos abordados.
- Resumo dos conceitos principais.



AVALIAÇÃO

- Ficha de avaliação de aprendizagens.
- Duração (50 minutos).



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

1 Qual das seguintes opções define programação modular? 10 pont

10 pont

- Uma técnica para escrever programas sem a utilização de funções.
- Uma forma de criar programas sem erros.
- Uma técnica usada para programar apenas em linguagens orientadas a objetos.
- Uma técnica que divide um programa em partes menores, chamados módulos ou subprogramas.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

2 Qual das seguintes opções **NÃO** é uma vantagem da programação modular?

- Facilitar a manutenção do código.
- Aumentar a redundância do código.
- Permitir o trabalho de equipa.
- Reutilizar código noutros outros projetos.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

3. Qual das seguintes instruções é considerada uma *função de "Output"*?

- printf().**
- main().**
- start().**
- scanf().**

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

4. Qual a etapa em que a solução é traduzida para *linguagem máquina*?

- Edição.
- Execução.
- Compilação.
- Pré-processamento.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

5. O que acontece durante o *pré-processamento*? 10 pontos

- Escrevemos o código num editor de texto.
- O código é preparado para compilação, analisando diretivas como **#include**.
- O programa é testado no ambiente de execução.
- As funções são otimizadas.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

6. Qual das seguintes afirmações sobre as *funções* é verdadeira?

- Um bloco de código que realiza uma tarefa específica.
- Uma **void** deve sempre retornar um valor.
- Todas as funções de um programa são executadas automaticamente.
- Uma função não pode chamar outra.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

7. Qual é a principal diferença entre uma *função void* e uma *função* com **return**?

- A **void** retorna valores, mas a com **return** não.
- A com **return** devolve valores, mas a **void** não.
- A **void** é sempre obrigatória em qualquer programa.
- A com **return** não pode ter argumentos.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

8. Qual das seguintes afirmações referente a variáveis locais é verdadeira?

- São declaradas dentro de uma função e só podem ser usadas pela mesma.
- São declaradas fora de qualquer função.
- Podem ser usadas em qualquer parte do programa.
- São inicializadas automaticamente com o valor 0.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

9. O que são *argumentos* de uma função?

- Instruções dentro do corpo da função.
- Variáveis locais usadas pelo programa.
- Variáveis que recebem valores e que são enviados para a função.
- Diretivas ao pré-processador incluídas no código.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

10. Qual das seguintes opções é uma **vantagem** da programação modular?

- Permite executar várias funções simultaneamente.
- Organiza o código e evitam a redundância.
- Complica a solução.
- Torna desnecessário a utilização da função main().

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

11. Qual das seguintes opções relativamente a tipo de dados primitivos *é verdadeira*?

- O *int* contém números reais.
- O *float* só contém números inteiro.
- O *double* admite valores reais com precisão.
- O *char* admite dados lógicos.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

12. Qual das seguintes afirmações relativamente a *funções* é verdadeira?

- Devolve mais do que um resultado em simultâneo.
- É sempre mais complexa.
- Permite realizar uma tarefa específica com retorno de um valor.
- Não necessita de cabeçalho.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

13. Qual das seguintes afirmações relativamente à etapa da compilação é **verdadeira**?

- Corrige automaticamente todos os erros do programa.
- Apenas traduz o código fonte para o código máquina.
- Identifica erros de sintaxe e de semântica no código fonte.
- Dá instruções ao pré-processador relativamente ao **#include**.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

14. Qual das seguintes afirmações **relativamente a subprogramas** é verdadeira?

- Não obedece às regras da programação estruturada.
- Pode ou não retornar valores.
- É um bloco de código que apenas serve para fazer inputs e outputs.
- Nunca precisa de ser chamado para ser executado.

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 1

15. Qual das seguintes opções contém instruções associadas à biblioteca **stdio.h**?

- while.
- scanf() e printf().
- sqrt().
- round().

Resposta:



CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 2

1. Qual o nome da função criada pelo programador no presente código?

Resposta:

```
1 # include <stdio.h>
2
3 int soma(int a, int b)
4 {
5     return a + b;
6 }
7
8 int main()
9 {
10     printf("%d\n", soma(3,5));
11     return 0;
12 }
```

CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 2

2. Qual o papel da instrução `return a + b`?

Resposta:

```
1  # include <stdio.h>
2
3  int soma(int a, int b)
4  {
5      return a + b;
6  }
7
8  int main()
9  {
10     printf("%d\n", soma(3,5));
11     return 0;
12 }
```

CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 2

3. Na função main() qual o valor devolvido pela função soma?

Resposta:

```
1  # include <stdio.h>
2
3  int soma(int a, int b)
4  {
5      return a + b;
6  }
7
8  int main()
9  {
10     printf("%d\n", soma(3,5));
11     return 0;
12 }
```

CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 2

4. Quais são as variáveis declaradas no código acima?

Resposta:

```
1 # include <stdio.h>
2
3 float CalculaMedia (float nota1, float nota2, float nota3)
4 {
5
6     return (nota1 + nota2 + nota3) /3.0;
7
8 }
9
10 int main()
11 {
12
13     float nota1, nota2, nota3, media;
14
15     printf("Insira a nota 1:");
16     scanf("%f", &nota1);
17
18     printf("Insira a nota 2:");
19     scanf("%f", &nota2);
20
21     printf("Insira a nota 3:");
22     scanf("%f", &nota3);
23
24     media = CalculaMedia(nota1, nota2, nota3);
25     printf("A media das 3 notas é %.2f", media);
26
27     return 0;
28
29 }
```

CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 2

5. Porque motivo neste código se declaram variáveis com o tipo primitivo **float**?

Resposta:

```
1 # include <stdio.h>
2
3 float CalculaMedia (float nota1, float nota2, float nota3)
4 {
5     return (nota1 + nota2 + nota3) /3.0;
6 }
7
8
9
10 int main()
11 {
12
13     float nota1, nota2, nota3, media;
14
15     printf("Insira a nota 1:");
16     scanf("%f", &nota1);
17
18     printf("Insira a nota 2:");
19     scanf("%f", &nota2);
20
21     printf("Insira a nota 3:");
22     scanf("%f", &nota3);
23
24     media = CalculaMedia(nota1, nota2, nota3);
25     printf("A media das 3 notas é %.2f", media);
26
27     return 0;
28
29 }
```

CORREÇÃO DA FICHA DE AVALIAÇÃO

GRUPO 2

6. Quantos valores recebe a *função*? Justifique a sua resposta.

Resposta:

```
1 # include <stdio.h>
2
3 float CalculaMedia (float nota1, float nota2, float nota3)
4 {
5     return (nota1 + nota2 + nota3) /3.0;
6 }
7
8
9
10 int main()
11 {
12     float nota1, nota2, nota3, media;
13
14     printf("Insira a nota 1:");
15     scanf("%f", &nota1);
16
17     printf("Insira a nota 2:");
18     scanf("%f", &nota2);
19
20     printf("Insira a nota 3:");
21     scanf("%f", &nota3);
22
23     media = CalculaMedia(nota1, nota2, nota3);
24     printf("A media das 3 notas é %.2f", media);
25
26     return 0;
27 }
28
29
```

RESUMO

- **Programação Modular:**

- Divisão de programas em módulos ou subprogramas para melhorar a organização e reutilização de código.

- **Vantagens:**

- Legibilidade.
- Reutilização de código.
- Facilidade na depuração e manutenção.
- Trabalho em equipa eficiente.

- **Conceito de Função:**

- Bloco de código reutilizável que realiza uma tarefa específica.
- Central na programação modular, permite agrupar instruções e dividir o programa em partes menores.

RESUMO

Estrutura de uma Função:

- **Cabeçalho:** Define o tipo de retorno, nome da função e argumentos.
- **Corpo:** Contém as instruções que a função executa.

• Exemplo:

```
int soma(int a, int b) {  
    return a + b;  
}
```

• Tipos de Funções:

- **Com retorno:** Retornam um valor para o programa principal.
- **Void:** Não retornam valor, mas realizam ações (ex.: exibem mensagens).
- **Argumentos:**
 - **Variáveis** locais da função que recebem valores na chamada.

RESUMO

- **Ciclo de Desenvolvimento de um Programa:**

- **Etapas:**

- Edição: Escrever o código.
- Pré-processamento: Preparar o código para compilação.
- Compilação: Transformar código-fonte em linguagem de máquina.
- Execução: Testar e validar o programa.

- **A Função main():**

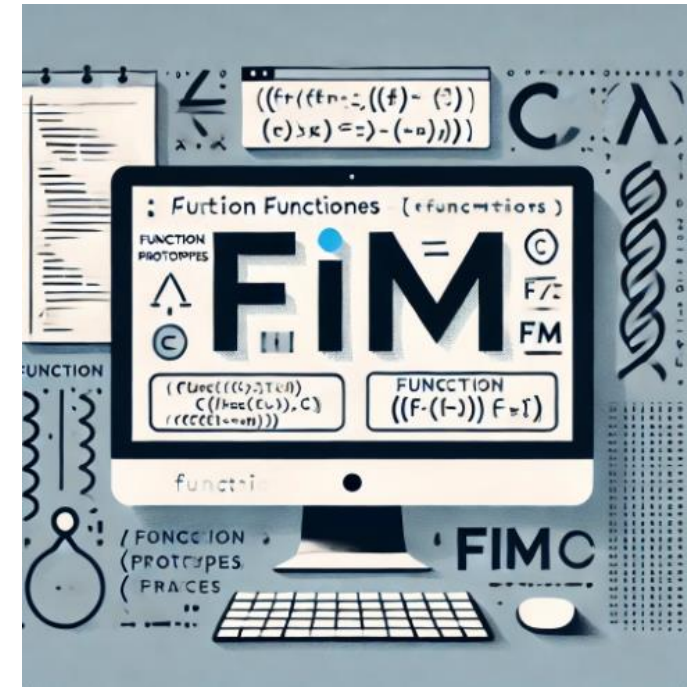
- Ponto de entrada de qualquer programa em C.

- **Estrutura básica:**

```
int main() {  
    printf("Olá, Mundo!\n");  
    return 0;  
}
```

RESUMO

- **Vantagens das Funções:**
 - Reaproveitamento e organização do código.
 - Evita repetição de blocos.
 - Facilita a leitura e separação de tarefas.



ESCOLA SECUNDÁRIA DE SANTO ANDRÉ

CURSO PROFISSIONAL TÉCNICO DE INFORMÁTICA - SISTEMAS

Fontes Bibliográficas

- Damas, L. (2019). *Linguagem C* (24ª ed.). Edições FCA.
- Deitel, H. M., & Deitel, P. J. (2015). *C: How to program* (8ª ed.). Pearson Education.
- Kernighan, B. W., & Ritchie, D. M. (1988). *The C programming language* (2ª ed.). Prentice Hall.
- King, K. N. (2008). *C programming: A modern approach* (2ª ed.). W. W. Norton & Company.
- Liberty, J. (2014). *Programming in C: A complete introduction to the C programming language*. Sams Publishing.
- Ponti Jr., M. P. (2011). *Uma breve introdução à criação de bibliotecas e makefiles em C/C++*. Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação.
- Sentance, S., Barendsen, E., Howard, N. R., & Schulte, C. (2023). *Computer science education: Perspectives on teaching and learning in school*. Bloomsbury Academic.

