**Logotipo, nome da empresa

Descrição gerada automaticamenteFACULDADE ANHANGUERA – UNIDADE BETIM**

**Curso:** Engenharia Civil & Engenharia Mecânica

**Disciplina:** Algoritmos e Lógica de Programação

**Professor:** Cristiano M. Nunes

**Período letivo:** 2.024\2

**# Aula 09 - Linguagem de Programação C:**

1. **Compilador On-line:** 
   1. <https://www.onlinegdb.com/online_c_compiler>
2. **Compilador DevC++:**
   1. Barra de Tarefas do Windows / Iniciar...
   2. Pasta BloodShedDev-C++ / Dev-C++
   3. Menu Arquivo / Novo / Arquivo Fonte
   4. Acesse Menu Arquivo / Salva Como...
   5. Nome do arquivo: informe o nome do arquivo
   6. Tipo do arquivo: C source (\*.c)
3. **Estrutura de um Programa na Linguagem C:**
   1. **Cabeçalho do Programa:**
      1. **Bibliotecas:**

#include <stdio.h>

#include <stdlib.h>

* 1. **Corpo do Programa:**
     1. **Método main():**

main( ) {

***< Código Fonte do Programa >;***

}

**Interface gráfica do usuário, Texto

Descrição gerada automaticamente**

1. **Bibliotecas básicas da Linguagem de Programação C:**
   1. #include <stdio.h>
      1. Esta biblioteca contém as definições de [subrotinas](https://pt.wikipedia.org/wiki/Sub-rotina) relativas às operações de [entrada/saída](https://pt.wikipedia.org/wiki/Entrada/sa%C3%ADda), como leitura de [dados](https://pt.wikipedia.org/wiki/Dados) digitados no [teclado](https://pt.wikipedia.org/wiki/Teclado_(inform%C3%A1tica)) e exibição de informações na [tela](https://pt.wikipedia.org/wiki/Tela) do [programa de computador](https://pt.wikipedia.org/wiki/Programa_de_computador).
   2. #include <stdlib.h>
      1. Esta biblioteca contém as definições de [sub-rotinas](https://pt.wikipedia.org/wiki/Sub-rotina) relativas às operações envolvendo alocação de memória, controle de processos, conversões de dados.
   3. #include <locale.h>
      1. Esta biblioteca contém as definições de [subrotinas](https://pt.wikipedia.org/wiki/Sub-rotina" \o "Sub-rotina) para a definição do idioma, acentuação gráfica e caracteres especiais.
   4. #include <math.h>
      1. Esta biblioteca contém as definições de [subrotinas](https://pt.wikipedia.org/wiki/Sub-rotina" \o "Sub-rotina) para a disponibilização das principais funções matemáticas.
2. **Tipo de dado na Linguagem de Programação C:**
   * bool
   * char
   * double
   * float
   * int
3. **Declaração de uma Variável Simples:**
   1. **Sintaxe:**
      * Tipo de dado <nome da variável>;
      * bool statusMatricula;
      * char sexo;
      * double salario;
      * float preco;
      * int idade;
4. **Declaração de uma Variável Constante:**
   1. **Sintaxe:**
      * **const** Tipo de dado<nome da variável>;
      * **const** double pi = 3.1416;
      * **const** float velocidadeGravidade = 9.8;
      * **const** int pesoProva1 = 4;
5. **Atribuição de valor na Variável:**
   1. **Sintaxe 01:**
      * bool statusMatricula;
      * statusMatricula = 1;
      * char sexo;
      * sexo = 'F';
      * int idade;
      * idade = 51;
   2. **Sintaxe 02:**
      * double salario = 1045.99;
      * float preco = 2.19;
      * int idade = 23;
6. **Comando de Saída de dados printf():**
   1. **Sintaxe 01:**
      * printf("Disciplina de Algoritmos Estruturados 2.024");
      * printf("Betim, 12 de Agosto de 2.024");
   2. **Sintaxe 02:**
      1. char sexo = 'F';
      2. printf("Sexo: %c", sexo);
      3. int idadeAluno = 23;
      4. printf("A idade do estudante é: %d", idadeAluno);
      5. float precoCombustivel = 2.19;
      6. printf("O preço do etatanol é: R$ %f", precoCombustivel);
      7. double salarioMinimo = 1045.99;
      8. printf("O valor do salário mínimo é: R$ %f", salarioMinimo);
7. **Comando de Entrada de dados scanf():**
   1. **Sintaxe 01:**
      1. int idadeAluno;
      2. printf("Quantos anos você tem?");
      3. scanf("%d",&idadeAluno);
   2. **Sintaxe 02:**
8. float precoProduto;
9. printf("Qual é o preço do produto?");
10. scanf("%f",& precoProduto);
11. **Operadores Aritméticos:**
    1. Adição: +
    2. Subtração: -
    3. Multiplicação: \*
    4. Divisão: /
12. **Operadores Aritméticos de Atribuição:**
    1. Adição: +=
    2. Subtração: -=
    3. Multiplicação: \*=
    4. Divisão: /=
    5. **Sintaxe:**
       1. int valorNumero = 20;
       2. valorNumero += valorNumero;
       3. float valorProva = 10.50;
       4. valorProva \*= 10;
13. **Operadores Aritméticos de Incremento/Decremento:**
    1. **Sintaxe:**
       1. int valorNumero = 2023;
       2. valorNumero++;
       3. float valorProva = 5.25;
       4. valorProva--;
14. **Operadores Relacionais:**
    * Igualdade: =
    * Menor que: <
    * Maior que: >
    * Menor ou igual que: <=
    * Maior ou igual que: >=
    * Diferente: !=
15. **Operadores Lógicos:**
    * AND (e): &&
    * OR (ou): ||
    * NOT (não): !
16. **Estrutura Sequencial:**
    * 1. [Exemplo00.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo00.c)
      2. [Exemplo01.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo01.c)
      3. [Exemplo02.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo02.c)
17. **Estrutura Condicional Simples:**
    * 1. [Exemplo03.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo03.c)
      2. [Exemplo04.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo04.c)
      3. [Exemplo05.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo05.c)
18. **Estrutura Condicional Composta:**
    * 1. [Exemplo06.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo06.c)
      2. [Exemplo07.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo07.c)
      3. [Exemplo08.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo08.c)
19. **Estrutura Condicional Aninhada:**
    * 1. [Exemplo09.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo09.c)
      2. [Exemplo10.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo10.c)
      3. [Exemplo11.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo11.c)
20. **Estrutura Condicional de Seleção:**
    * 1. [Exemplo12.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo12.c)
      2. [Exemplo13.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo13.c)
21. **Estrutura Iteração:**
    1. **Introdução:**
       1. Na linguagem C, e em todas as outras linguagens modernas de programação, comandos de iteração (também chamados laços de repetição) permitem que um conjunto de instruções seja executado até que ocorra uma certa condição. Essa condição pode ser predefinida (como no laço FOR) ou com o final em aberto (como nos laços while e do-while).
22. **Estrutura Iteração *“for”*:**
    1. **Introdução:**
       1. Na estrutura de iteração *“for”* a condição de iteração (repetição) é definida e verificada no início da estrutura. Quando utilizamos esta estrutura sabemos exatamente o número de repetição das instruções.
    2. **Sintaxe:**
       1. *for(inicialização; condição; incremento/decremento)*

*{ bloco de instruções; }*

* + 1. inicialização: contador de repetição;

condição: condição de repetição;

incremento/decremento: ++ ou –

* + 1. [Exemplo14.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo14.cpp)
    2. [Exemplo15.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo15.cpp)
    3. [Exemplo16.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo16.cpp)

1. **Estrutura Iteração *“while”*:**
   1. **Introdução:**
      1. Na estrutura de iteração *“while”* a condição de iteração (repetição) é definida e verificada no início da estrutura. A condição de repetição pode ser qualquer expressão lógica que assuma um valor verdadeiro. Quando utilizamos esta estrutura não sabemos exatamente o número de repetição das instruções, mas que as instruções serão repetidas conforme a condição determinada.
   2. **Sintaxe:**
      1. *while(condição)*

*{ bloco de instruções;*

*incremento/decremento; }*

* + 1. [Exemplo17.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo17.cpp)
    2. [Exemplo18.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo18.cpp)
    3. [Exemplo19.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo19.cpp)

1. **Estrutura Iteração *“do-while”*:**
   1. **Introdução:**
      1. Na estrutura de iteração *“do-while”* a condição de iteração (repetição) é definida e verificada no final da estrutura. A condição de repetição pode ser qualquer expressão lógica que assuma um valor verdadeiro. Quando utilizamos esta estrutura sabemos que as instruções serão executadas no mínimo uma vez.
   2. **Sintaxe:**
      1. *do{*

*bloco de instruções;*

*incremento/decremento;*

*} while (codição);*

* + 1. [Exemplo20.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo20.cpp)
    2. [Exemplo21.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo21.cpp)
    3. [Exemplo22.c](file:///G:\Root\20201\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2010\Exemplo22.cpp)

1. **Estrutura de Armazenamento Array:**
   1. **Introdução:**
      1. Um array é uma coleção de variáveis do mesmo tipo que é referenciada por um nome em comum. Um elemento específico em um array é acessado por meio de um índice, ou seja, a sua posição no array. Na linguagem C, todos os array’s consistem de posições contíguas de memória. O array são classificados em array unidimensional (vetor), array bidimensional (matriz) e array multidimensional (matriz AxBxC).
   2. **Sintaxe Vetor**
      1. tipo de dado nome do array [tamanho];
      2. float vetorNota[4];
      3. [Exemplo23.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo23.cpp)
      4. [Exemplo24.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo24.cpp)
      5. [Exemplo25.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo25.cpp)
      6. [Exemplo26.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo26.cpp)
      7. [Exemplo27.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo27.cpp)
      8. [Exemplo28.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo28.cpp)
      9. [Exemplo29.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo29.cpp)
      10. [Exemplo30.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo30.cpp)
   3. **Sintaxe Matriz**
      1. tipo de dado nome do array [dimensao][dimensao];
      2. float matrizNota[2][2];
      3. [Exemplo31.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo23.cpp)
      4. [Exemplo32.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo24.cpp)
      5. [Exemplo33.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo25.cpp)
      6. [Exemplo34.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo26.cpp)
      7. [Exemplo35.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo27.cpp)
      8. [Exemplo36.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo28.cpp)
      9. [Exemplo37.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo29.cpp)
      10. [Exemplo38.c](file:///E:\Root\2023-1\06%20-%20Algoritmos%20e%20Lógica%20de%20Programação\Algoritmos%20e%20Lógica%20de%20Programação\Aula%2012\Exemplo30.cpp)