

1. Declarar e Inicializar Variáveis de Diferentes Tipos de Dados

```
#include <stdio.h>
```

```
int main() {  
    int intVar = 10;  
    double doubleVar = 20.5;  
    float floatVar = 30.5f;  
    char charVar = 'A';  
  
    printf("Valor de intVar: %d\n", intVar);  
    printf("Valor de doubleVar: %.2lf\n", doubleVar);  
    printf("Valor de floatVar: %.2f\n", floatVar);  
    printf("Valor de charVar: %c\n", charVar);  
  
    return 0;  
}
```

2. Declarar, Inicializar e Alterar o Valor de uma Variável

```
#include <stdio.h>
```

```
int main() {  
    int var = 100;  
    printf("Valor inicial de var: %d\n", var);  
  
    var = 200;  
    printf("Novo valor de var: %d\n", var);  
  
    return 0;  
}
```

3. Definir Constantes com #define e const

```
#include <stdio.h>
```

```
#define PI 3.14159
```

```
int main() {  
    const int MAX_VALUE = 100;  
  
    printf("Valor da constante PI (definida com #define): %.5f\n", PI);  
    printf("Valor da constante MAX_VALUE (definida com const): %d\n", MAX_VALUE);  
}
```

```
// Tentativa de alteração das constantes (comentadas para evitar erro de compilação)
// PI = 3.14; // Isso causará um erro de compilação
// MAX_VALUE = 200; // Isso causará um erro de compilação

return 0;
}
```

Resposta às Perguntas sobre Constantes

O que acontece se você tentar alterar o valor de uma constante definida como `const`?

Se você tentar alterar o valor de uma constante definida como `const`, o compilador emitirá um erro, pois as constantes não podem ser modificadas após a sua definição.

O que acontece se você tentar alterar o valor de uma constante definida como `#define`?

Se você tentar alterar o valor de uma constante definida com `#define`, você não receberá um erro diretamente porque `#define` é uma diretiva de pré-processador. No entanto, a tentativa de alteração será ignorada, pois `#define` substitui texto antes da compilação e não cria uma variável que possa ser modificada.

4. Declaração de Variáveis e Constantes com Comentários

```
#include <stdio.h>
```

```
#define GRAVITY 9.81 // Definindo uma constante com #define
#define MAX_USERS 100 // Definindo uma constante com #define
```

```
int main() {
    // Variáveis
    int age = 25; // Declarando e inicializando uma variável inteira
    float height = 1.75f; // Declarando e inicializando uma variável de ponto flutuante
    double balance = 12345.67; // Declarando e inicializando uma variável de ponto flutuante
    de precisão dupla
    char grade = 'A'; // Declarando e inicializando uma variável de caractere
```

```
    // Constantes
```

```
    const double PI = 3.14159; // Definindo uma constante com const
    const int MAX_SCORE = 100; // Definindo uma constante com const
    const float TEMPERATURE = 36.6f; // Definindo uma constante com const
    const char INITIAL = 'J'; // Definindo uma constante com const
```

```
    // Imprimindo as variáveis
```

```
    printf("Idade: %d\n", age); // Imprime a idade
    printf("Altura: %.2f\n", height); // Imprime a altura
    printf("Saldo: %.2lf\n", balance); // Imprime o saldo
    printf("Nota: %c\n", grade); // Imprime a nota
```

```
    // Imprimindo as constantes
```

```
    printf("Gravidade: %.2f\n", GRAVITY); // Imprime a gravidade
    printf("Máximo de usuários: %d\n", MAX_USERS); // Imprime o máximo de usuários
```

```
printf("Valor de PI: %.5f\n", PI); // Imprime o valor de PI
printf("Pontuação máxima: %d\n", MAX_SCORE); // Imprime a pontuação máxima
printf("Temperatura: %.1f\n", TEMPERATURE); // Imprime a temperatura
printf("Inicial: %c\n", INITIAL); // Imprime a inicial

return 0;
}
```

Este programa demonstra a declaração e inicialização de variáveis e constantes, além de comentários explicando cada linha de código.