



INSTITUTO FEDERAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE POO e LAB. POO
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Atividade Unidade I - 2 - Fundamentos C++

Instruções

Responda às questões teóricas abaixo diretamente neste documento. As questões práticas devem ser anexadas separadamente em arquivos .cpp ou .zip. As questões de 1 a 7 valem 2. A questão 8 vale 3.

Questões

1. Determine o tipo deduzido em cada uma das definições **auto**.

```
int i = 10;  
double d = 1.0;  
auto ai = i; R: Tipo inteiro  
const auto cai = i; R: Tipo constante inteiro  
auto r1 = i + 100L; R: Tipo long int  
auto r2 = i + d; R: Tipo double  
const auto r3 = d + 3.0L; R: Tipo constante long double
```

2. Quais os valores de i e d após cada atribuição?

Considere:

```
int i;  
double d;
```

- a. `d = i = 3.5;` R: A variável d é igual a 3.0, e a variável i é igual a 3
 - b. `i = d = 3.5;` R: A variável i é igual a 3, e a variável d é igual a 3.5
3. Qual o valor de i e j após execução no código abaixo?

```
int i = 0, j = 0;  
j = (i += 3, i + 3);  
Valor de j = 6, Valor de i = 3.
```

4. O trecho de código a seguir não compila por um erro de precedência de operadores. Explique o que está acontecendo e corrija o código.

```
string s = "word";  
string p1 = s + s[s.size() - 1] == 's' ? "" : "s";
```

Correção:

```
std::string p1 = s + ((s[s.size()-1] == 's') ? "" : "s");
```

Explicação: O motivo do erro seria a falta de parênteses para se referir às expressões, tendo em vista que possui duas expressões na linha de código.

5. Qual o resultado do trecho abaixo? Explique.

```
int i = 5, j = 2;  
double d = static_cast<double>(i/j);
```

R: Da maneira que o código se encontra, o resultado retornará 2, pois ele não faz o static da maneira correta, para que retorne 2.5, é necessário envolver o i em parênteses para que dessa forma ele se torne uma variável double

6. Escreva um programa que lê uma string (com espaços) do usuário e apresente na tela a string com os sinais de pontuação removidos (incluindo os espaços).

Código:

```

C++ testes.cpp > ...
1  ✓ #include <iostream>
2    #include <ctype.h>
3
4  ✓ int main(int argc, char const *argv[])
5  {
6      std::string frase = "";
7      std::string frasereformulada = "";
8
9      std::cout << "Digite a frase: ";
10     std::getline(std::cin, frase);
11
12     int percorrer;
13
14     for(percorrer = 0;percorrer < frase.size(); percorrer++){
15         if ((isspace(frase[percorrer])) || (ispunct(frase[percorrer]))){
16             continue;
17         }
18         frasereformulada = frasereformulada + frase[percorrer];
19     }
20
21     std::cout << frasereformulada;
22
23     return 0;
24 }
25

```

7. Os motoristas se preocupam com o consumo de combustível dos seus veículos. Um motorista monitorou vários tanques cheios de gasolina registrando a quilometragem dirigida e a quantidade de combustível em litros utilizados para cada tanque cheio. Desenvolva um programa C++ para inserir os quilômetros percorridos e a quantidade de litros de gasolina utilizados para cada tanque. O programa deve calcular e exibir o consumo em quilômetros/litro para cada tanque cheio, bem como a soma total de litros consumidos em cada ponto. O programa deve rodar enquanto a quilometragem lida for diferente de -1.

Código:

```

1  #include <iostream>
2  #include <string>
3  ▼ int main() {
4
5      float quilometragem = 0.f, combustivel = 0.f, somacombustivel =
6      0.f;
7  ▼ while (1){
8
9      std::cout << "Digite a quilômetragem percorrida: ";
10     std::cin >> quilometragem;
11  ▼     if (quilometragem == -1.0){
12         break;
13     }
14
15     std::cout << "Digite a quantidade de combustivel: ";
16     std::cin >> combustivel;
17     somacombustivel += combustivel;
18
19     std::cout << std::endl << "Consumo de quilômetros por litro: "
20     << quilometragem/combustivel << std::endl << std::endl;
21 }
22
23     std::cout << "A soma total de litros de combustível consumidos é
24     igual a : " << somacombustivel << "L";|
25
26     return 0;
27 }

```

8. Escreva um programa que utiliza instruções for para imprimir cada um dos seguintes padrões separadamente. Utilize loops for para gerar os padrões. Todos os asteriscos (*) devem ser impressos por uma única instrução na forma `cout << '*'`;

a.

*

**

```

1  #include <iostream>
2  #include <string>
3  int main() {
4
5      for (auto cont = 1; cont < 6; cont++){
6          std::string repeticao(cont, '*');
7          std::cout << repeticao << std::endl;
8      }
9      return 0;
10 }

```

b.

**

*

```

#include <iostream>

int main(int argc, char const *argv[])
{
    for (auto contador = 5; contador > 0; contador--){
        int espacos = 5 - contador;
        std::string espacamento(espacos, ' ');
        std::string repeticao(contador, '*');
        std::cout << espacamento;
        std::cout << repeticao << std::endl;
    }
    return 0;
}

```