



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 - 3 - Fundamentos C++

Instruções

Responda às questões abaixo. Questões teóricas devem ser respondidas aqui. Questões práticas exigem que a resposta seja anexada em um arquivo .cpp separadamente.

Dica: use uma IDE de preferência para realizar experimentos, fazer análises e observações antes de responder às questões. Não é exigido que o aluno decore as nuances e especificidades da linguagem, mas que compreenda e tenha a capacidade de discerni-las.

A questão 1 vale um ponto, enquanto as demais valem 2 pontos.

Questões

1. Explique a diferença entre parâmetro e argumento.

Resposta:

Parâmetros são as variáveis que são passadas para uma função em seu cabeçalho

Já os argumentos são a inicialização dos parâmetros.

2. Explique a diferença entre uma variável local e uma variável local static.

Resposta:

Variáveis locais são variáveis que, quando definidas dentro de um escopo de uma função, elas existirão apenas no escopo da função.

Já as variáveis locais static, são variáveis que mesmo criadas dentro do escopo de uma função, elas permanecem existentes até o fim do programa.

3. Escreva uma função que retorna 0 (zero) quando chamada pela primeira vez e então gera números em sequência em cada uma das chamadas posteriores.

Código:

```

1  ✓ #include <iostream>
2    #include <ctype.h>
3
4    unsigned posterior(void);
5
6  ✓ int main(int argc, char const *argv[])
7    {
8
9  ✓    for (auto cont = 0; cont < 10; cont++){
10       std::cout << posterior() << " ";
11    }
12
13
14    return 0;
15  }
16
17  ✓ unsigned posterior(void){
18
19    static unsigned numero = 0;
20    return numero++;
21  }

```

4. Usando ponteiros, escreva uma função chamada **swap** que troca os valores dos seus 2 parâmetros.

Código:

```

1 #include <iostream>
2 #include <string>
3
4 void troca(int *valor1, int *valor2);
5
6 int main(int argc, char const *argv[])
7 {
8     int valor1 = 100, valor2 = 200;
9     std::cout << "ANTES DA TROCA : " << std::endl;
10    std::cout << "Valor 1: " << valor1 << " Valor 2: " << valor2 << std::endl;
11    std::cout << "DEPOIS DA TROCA : " << std::endl;
12    troca(&valor1, &valor2);
13    return 0;
14 }
15
16
17 void troca(int *valor1, int *valor2){
18     int *auxiliar = valor1;
19     valor1 = valor2;
20     valor2 = auxiliar;
21     std::cout << "Valor 1: " << *valor1 << " Valor 2: " << *valor2;
22 }

```

5. Usando referências, escreva uma função chamada **swap** que troca os valores dos seus 2 parâmetros.

Código:

```

1 #include <iostream>
2 #include <string>
3
4 void troca(int &valor1, int &valor2);
5
6 int main(int argc, char const *argv[])
7 {
8     int valor1 = 100, valor2 = 200;
9     std::cout << "ANTES DA TROCA : " << std::endl;
10    std::cout << "Valor 1: " << valor1 << " Valor 2: " << valor2 << std::endl;
11    std::cout << "DEPOIS DA TROCA : " << std::endl;
12    troca(valor1, valor2);
13
14    return 0;
15 }
16
17
18 void troca(int &ref1, int &ref2){
19     int auxiliar = ref1;
20     ref1 = ref2;
21     ref2 = auxiliar;
22     std::cout << "Valor 1: " << ref1 << " Valor 2: " << ref2 << std::endl;
23 }

```

6. Escreva uma função que determina se uma string C++ contém alguma letra maiúscula.

Código:

```

1 #include <iostream>
2 #include <string>
3
4 int verificador(const std::string &str);
5
6 int main(int argc, char const *argv[]){
7     std::string str;
8     std::cout << "Digite a string: ";
9     std::getline(std::cin, str);
10    std::string texto = (verificador(str)) ? "Tem Maiuscula": "Nao tem maiuscula";
11    std::cout << texto;
12    return 0;
13 }
14
15 int verificador(const std::string &str){
16     static int posicao = 0;
17     while(str[posicao]){
18         if(isupper(str[posicao])){
19             return 1;
20         }
21         posicao++;
22     }
23     return 0;
24 }

```

7. Escreva uma função que receba uma string C++ e converta cada letra para minúscula.

Código:

```

1 #include <iostream>
2 #include <string>
3
4 void colocarmaiuscula(std::string &str);
5
6 int main(int argc, char const *argv[]){
7     std::string str;
8     std::cout << "Digite a string: ";
9     std::getline(std::cin, str);
10    colocarmaiuscula(str);
11    std::cout << str;
12    return 0;
13 }
14
15
16 void colocarmaiuscula(std::string &str){
17     static int posicao = 0;
18     while(str[posicao]) {
19         str[posicao] = toupper(str[posicao]);
20         posicao++;
21     }
22 }

```

8. De acordo com o trabalhado em sala, a definição dos parâmetros usados nas questões 6 e 7 devem ser iguais? Qual seria a diferença e por quê?

Resposta: Não, pois a melhor maneira de definir os parâmetros de ambas questões seria:

questão 6 - `const string &str`, pois será uma string que não será modificada

questão 7 - `string &str`, tendo em vista que ocorrerá uma mudança na própria string

9. Qual, caso haja, das seguintes declarações é um erro? Por quê?

a. `int ff(int a, int b = 0, int c = 0);`

Correto

b. `char *init(int ht = 24, int wd, char bckgrnd);`

Erro, pois o parâmetro que possui valor atribuído não está na direita, nem os demais parâmetros possuem atribuições.

10. Considerando a declaração abaixo, quais chamadas são legais e quais são ilegais (considerando a intenção do programador)? Por quê?

```
char *init( int ht, int wd = 80, char bckgrnd = ' ' );
```

a. `init();`

Erro, é necessário passar o valor do parâmetro ht

b. `init(24, 10);`

Correto, o parâmetro bckgrnd assumirá valor ' '

c. `init(14, '*');`

Correto, o parâmetro ht assumirá 14 e o wd assumirá valor 80

11. Analise o seguinte trecho de código e responda:

```
void print(const int ia[10])
{
    for (int i = 0; i != 10; i++)
        cout << ia[i] << endl;
}
```

Existe algum problema com o código? Que comportamento esperar? Se houver algum erro, corrija.

Resposta: a função print recebe como parâmetro uma lista int de tamanho 10, onde irá exibir na tela todo o conteúdo desta lista, através de um laço de repetição

12. Indique se a seguinte afirmação é Verdadeira ou Falsa: “Todos os argumentos para as chamadas de função C++ são passados por valor.”.

Resposta: Negativo, pois em c++, também é possível passar argumentos por referência.

13. O que faz o programa a seguir?

```
#include <iostream>

using std::cout, std::cin, std::endl;

int mystery(int, int);

int main()
{
    int x, y;

    cout << "Entre com dois inteiros: ";
    cin >> x >> y;
    cout << "Resultado: " << mystery(x, y) << endl;
    return 0;
}

//O parâmetro b deve ser um inteiro positivo
int mystery(int a, int b)
{
    if (b == 1)
        return a;
    else
        return a + mystery(a, b - 1);
}
```

Resposta:

O programa a seguir tem como objetivo mostrar na tela o resultado entre a multiplicação entre os inteiros a e b, de maneira recursiva, onde o retorno da função, sempre que b for diferente de 1, será a soma de A com ele mesmo ate que b se iguale a 1, através da subtração do mesmo.