

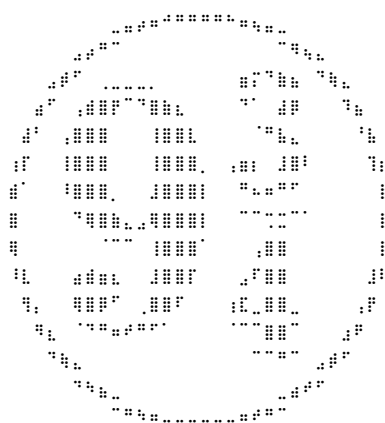
INSTITUTO FEDERAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE POO e LAB. POO
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Atividade Unidade I - 1 - Fundamentos C++

Instruções

Responda às questões abaixo. Pode usar este próprio documento. Questões práticas devem ser anexadas separadamente.

Uma aventura vai começar...



Legenda:

Quirrell: Questões mais simples e diretas

Olho-Tonto: Questões de nível intermediário

Dolores: Questões de nível mais difícil, exigentes

Voldemort: Questões com nível de exigência altíssimo, desafiadoras

Questões

Teóricas

1. **Quirrell** Para que servem os espaços de nome (Namespace) da C++?

R:

2. **Quirrell** Qual é a diferença, se é que existe, entre as seguintes definições:

```
int month{ 10 }, day{ 11 };
```

```
int month2{ 010 }, day2{ 011 };
```

```
int month3{ 0x10 }, day3{ 0x11 };
```

```
int month4{ 0b10 }, day4{ 0b11 };
```

3. **Quirrell** Qual a diferença entre definição e declaração de uma variável ou função em C++?

R:

4. **Quirrell** Vimos que a E/S da C++ se dá por meio dos objetos cout e cin. Para enviar um fluxo de dados para a saída usamos o operador << com cout, e para receber um fluxo de dados da entrada padrão usamos >> com cin. Originalmente, tais operadores são usados no contexto de números inteiros para realizar operações a nível de bit (bitwise), mas no contexto de E/S com cout e cin, eles mudam de comportamento. Como se chama essa técnica/recurso? E qual a finalidade desse recurso?

R:

5. **Olho-Tonto** O que é e qual é a diferença entre Ponteiros e Referências?

R:

6. **Olho-Tonto** Qual a diferença entre o array-C, o array-C++ e o vector?

R:

7. **Dolores** Quais das seguintes inicializações estão corretas? Quais estão incorretas? Indique o motivo em cada caso.

a. `int i{ -1 }, &r{ 0 };`

b. `const int i2{ i }, &r{ i };` //considere o i definido na letra a.

c. `int *const p2{ &i2 };` //considere o i2 definido na letra b.

d. `const int j{ -1 }, &s{ 0 };`

- e. `const int *p1{ &i2 }; //considere o i2 definido na letra b.`
- f. `const int *const p3{ &i2 }; //considere o i2 definido na letra b.`
- g. `const int &const r2{};`

8. **Olho-Tonto** Considere o código abaixo e responda:

```
int i{42};
int main()
{
    int i{100};
    int j{i};

    return 0;
}
```

- a. Existe algum problema (de compilação) com o código acima? Por quê?

R:

- b. Da forma como o código se encontra, qual será o valor de j?

R:

- c. Considerando que temos duas variáveis i em escopos diferentes, que modificação seria necessária no código para a variável j ser inicializada com o valor da variável i de escopo global.

R:

9. **Olho-Tonto** Qual a saída do código abaixo? Explique.

```
#include <iostream>
int main()
{
    int i, &ri{i};
    i = 5; ri = 10;
    std::cout << i << " " << ri << std::endl;
    return 0;
}
```

R:

10. **Olho-Tonto** Sabemos que a linguagem C++ usa alguns símbolos com diferentes significados/semânticas a depender do contexto. Isso posto, o que faz o seguinte trecho de código?

```
int i{10};
int *pi{&i};
*pi = *pi * *pi;
```

R:

11. **Olho-Tonto** Determine o tipo deduzido em cada uma das definições **auto**.

```
int i{10};
double d{1.0};
auto ai{i}; //R:
const auto cai{i}; //R:
auto r1{i + 100L}; //R:
auto r2{i + d}; //R:
const auto r3{d + 3.0L}; //R:
```

12. **Olho-Tonto** Quais os valores de i e d após cada atribuição?

Considere:

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

- a. d = i = 5.5;

R:

- b. i = d = 5.5;

R:

13. **Olho-Tonto** Qual o valor de i e j após execução no código abaixo?

```
int i{0}, j{0};
j = (i += 5, i + 5);
```

R:

14. **Olho-Tonto** O trecho de código a seguir não compila por um erro de precedência de operadores. O programador intencionou avaliar se a palavra armazenada na string **s** estaria no plural. Em caso afirmativo, a manteria assim, mas em caso negativo ele gostaria de deixá-la no plural. Explique o que está acontecendo e corrija o código.

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

R:

15. **Olho-Tonto** Para que serve o operador `static_cast` da C++?

R:

16. **Olho-Tonto** Qual o resultado do trecho abaixo? Explique.

```
int i{5}, j{2};
```

```
double d{ static_cast<double>(i/j) };
```

R:

17. **Quirrell** Explique a diferença entre parâmetro e argumento.

R:

18. **Quirrell** Por que é importante aplicarmos o qualificador **const** quando estivermos trabalhando com parâmetros de referência?

R:

19. **Quirrell** Qual a diferença entre uma variável local e uma variável local static?

R:

20. **Quirrell** Qual, caso haja, das seguintes declarações é um erro? Por quê?

- a. `int myFunc(int a, int b = 0, int c = 0);`
- b. `char *init(int ht = 24, int wd, char bckgrnd);`

R:

21. **Olho-Tonto** Considerando a declaração abaixo, quais chamadas são legais e quais são ilegais (considerando a intenção do programador)? Por quê?

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

- a. `init();`
- b. `init(24, 10);`
- c. `init(14, '*');`

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

R:

23. **Dolores** 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);
}

```

Práticas

24. **Olho-Tonto** Usando a biblioteca `iostream` da C++, escreva um programa que leia o primeiro nome e 3 notas de um aluno, e em seguida exiba na tela uma mensagem como "O aluno FULANO obteve média MÉDIA". Considere a média ponderada, sendo os pesos 3, 4 e 5, respectivamente, das notas 1, 2 e 3.
25. **Olho-Tonto** Com a escalada alarmante dos preços dos combustíveis, os motoristas têm se preocupado cada vez mais com o consumo de seus possantes. Por meio de um programa em C++, ajude-os a monitorar o consumo, registrando, a cada abastecimento, a quantidade de litros para encher o tanque e a quilometragem dirigida até ali. Seu programa deve calcular e exibir o consumo em quilômetros/litro para cada tanque cheio, bem como a soma total de litros consumidos. O programa deve rodar enquanto a quilometragem lida for diferente de -1.
26. **Olho-Tonto** Escreva uma função que retorna 0 (zero) quando chamada pela primeira vez e então gera números em sequência em cada uma das chamadas posteriores.
27. **Olho-Tonto** Usando **ponteiros**, escreva uma função chamada **swap** que troca os valores dos seus 2 parâmetros.
28. **Olho-Tonto** Usando **referências**, escreva uma função chamada **swap** que troca os valores dos seus 2 parâmetros.
29. **Olho-Tonto** Escreva uma **função** que determina se uma string C++ contém alguma letra maiúscula.
30. **Olho-Tonto** Escreva uma **função** que receba uma string C++ e converta cada letra para minúscula.
31. **Dolores** Escreva um **função** que determina se um array-C++ de inteiros recebido como referência contém algum número primo.

32. **Dolores** Escreva um **função** que determina se um vector recebido como referência contém algum número primo.
33. **Dolores** Qual a principal distinção entre o uso de array-C++ e vector como parâmetros? O parâmetro baseado em array-C++ serve para qualquer array-C++? E o baseado em vector? Explique.
34. **Dolores** Escreva um programa que utiliza instruções for para imprimir cada um dos seguintes padrões separadamente. Utilize loops for para gerar os padrões. Atenção à seguinte restrição: **cada asteriscos (*) deve ser impresso por uma única instrução na forma cout << “*”, isto é, devem ser impressos individualmente.**

a.

```
*  
**  
***  
****  
*****
```

b.

```
*****  
****  
***  
**  
*
```

c.

```
*  
***  
*****  
***  
*
```

d.

```
*
```

Voldemort está tramando algo para a próxima lista...

```

      -.,-----,
--  _.-.-.- / ' -.-      -.,-----,
|  \ -) _- ' -.- \      -.,-----, \)
/ _- ' -.- \ ' -.-, _- (c-(6)=(6)
, \ ' -.- \ ' -.-, ' >\ " )
: ; ; ; , ' -.- ' -.-' ( ( " / \ -=' /
: ; ; ; ; , ' -.- , \ -.-) ' -.-' -.-'
: ; ; ; ; ; , ' -.- / ' -.- | Y / _- / ' \
      ' ' ' ' ' F | _- / _- ' -.- \
      L \ \ / _- ' -.- \
      .-, -, - | \ ' -.-, \ _ _ |
// ' L / \ , ("---', =`)7
| \ -.- : _- , \ / ' \ -.- L, -' -.-
' -.-' ' -.- \ _- / _- L . ' ' . //
      [ ( /
      ) `{
      \ _-)
```