



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 diretamente neste documento. As questões valem 2 pontos, cada.

Questões

1. Qual é a diferença, se é que existe, entre as seguintes definições:

`int month = 10, day = 15;`

`int month = 010, day = 015;`

R: Na primeira definição, as variáveis `month` e `day` estão como números inteiros na base decimal, já na segunda definição, as mesmas variáveis estão como valores inteiros em base octal.

2. Qual a diferença entre definição e declaração de uma variável em C++?

R: É chamado de definição quando se cria/inicializa um objeto/variável no código, já declaração, é quando se comunica a existência de um objeto/variável.

3. 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.

R: Código:

```
#include <iostream>
```

```
#include <string>
```

```
//nomeando as funções que utilizaremos do std

using std::string;

using std::cout;

using std::cin;


int main(){


    //definição das variáveis

    float nota1 = 0.0f, nota2 = 0.0f, nota3 = 0.0f, media = 0.0f;

    string nome;


    //perguntando nome e notas ao usuário

    cout << "Digite seu primeiro nome: ";

    cin >> nome;


    cout << "Primeira nota: ";

    cin >> nota1;


    cout << "Segunda nota: ";

    cin >> nota2;


    cout << "Terceira nota: ";

    cin >> nota3;


    //cálculo para atribuir um valor a média0
```

```

media = (nota1*3 + nota2*4 + nota3*5)/12;

//imprimindo o resultado

cout << "O aluno " << nome << " obteve media = " << media;

return 0;
}

```

4. Vimos que a E/S da C++ se dá por meio dos objetos cout e cin. Para enviar um fluxo de caracteres para a saída usamos o operador << com cout e, para receber algum dado 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?

R: Esse recurso se chama sobrecarga, onde uma função/operador com semânticas iguais recebem funcionalidades diferentes, dependendo de qual escopo esteja sendo especificado, como exemplo do operador << e std::cout <<, onde possuem funcionalidades diferentes.

5. Considere o código abaixo e responda:

```

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

```

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

R: Não existe nenhum problema de compilação, tendo em vista que as variáveis "i" possuem duas classificações, uma como variável local e outra como variável global.

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

R: 100, pois ela recebe o valor da variável "i" mais próxima, que possui valor 100.

- c. Considerando que temos duas variáveis i, que modificação seria necessária no código para a variável j ser inicializada com o valor da outra variável i.

R: Para que “j” receba o valor da variável “i” global (42), é necessário realizar a seguinte modificação:

```
int j = ::i;
```

Dessa forma, “j” receberá o valor da variável global “i”.

6. O que é uma Referência em C++?

R: Uma referência, sem ser utilizada como ponteiro, seria um segundo nome para uma mesma variável.

7. 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: Na primeira linha, define-se duas variáveis, uma do tipo inteiro i, e outra inteira que é referência de i, ou seja, um segundo nome para uma mesma variável.

Na segunda linha, é declarado o valor 5 para i, em seguida, declara-se o valor 10 para ri, portanto, i também recebe o valor 10.

E por fim, é imprimido na tela o valor de i, que, devido ri (referência de i) ter recebido valor 10, será imprimido o valor 10 na tela.

8. Qual é a diferença entre Ponteiros e Referências?

R: Referência seria um segundo nome definido para uma outra variável já existente, em contrapartida, o ponteiro é uma variável que recebe o endereço de memória de uma variável.

9. O que faz o seguinte trecho de código?

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

R: Na primeira linha de código, foi definido uma variável inteira i que recebe o valor 10.

Na segunda linha de código, é criada a variável `pi`, um ponteiro que recebe o endereço de memória da variável `pi`.

Já na terceira linha de código, o conteúdo apontado por `pi` recebe ele mesmo vezes ele mesmo, ou seja, possuindo resultado 100.

10. Quais das seguintes inicializações estão corretas? Quais estão incorretas? Indique o motivo em cada caso.

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

R: A variável `i` está correta, porém a variável `&r` está incorreta, pois `10` é um valor `const int`, uma variável que é uma referência só pode receber outra variável como valor.

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

R: Ambas estão corretas, a variável `const int i2` recebe o valor de `i` (-1) e a variável referência `r` recebe o valor de `i` (-1).

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

R: Errado, a semântica do código está errada.

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

R: Corretas, a variável constante `int j` está recebendo um valor `int` constante, e a variável referência `s` também recebe um valor inteiro constante.

e. `const int *p1 = &i2; //considere o i2 definido na letra b.`

R: Correto, pois a variável constante inteira `p1`, que é um ponteiro, está apontando para uma variável que também é constante inteira.

f. `const int *const p3 = &i2; //considere o i2 definido na letra b.`

R: Correto, foi criada uma variável ponteiro constante, ou seja, que não pode mudar a variável para qual esteja apontando, do tipo `const int`.

g. `const int &const r2;`

R: Errado, pois a variável referência criada não está se referindo a ninguém.