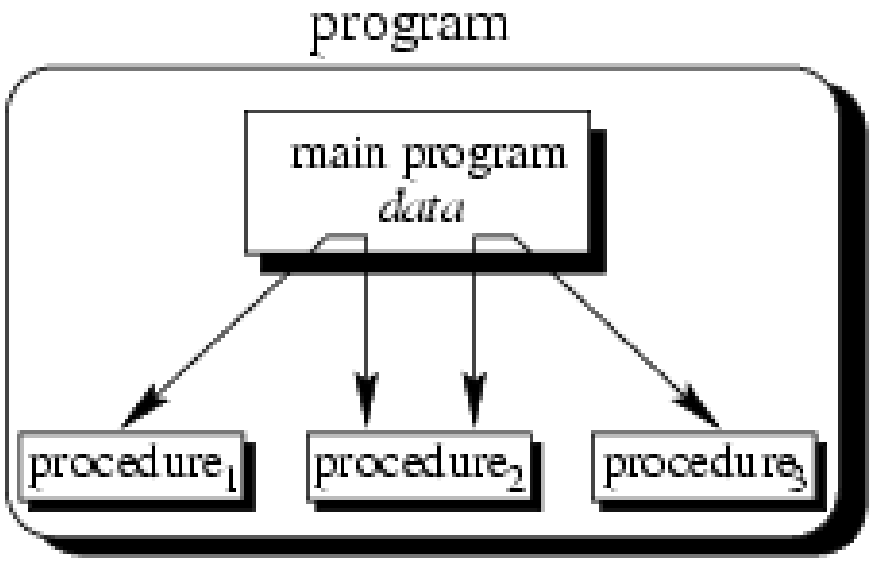
Programação em C ++

Prof. MSc. Rodrigo D. Malara
Universidade de Araraquara

Introdução ao C ++

- Conceito de programação
- C ++ básico
- C ++ é uma extensão de C

Linguagens Estruturadas (I)



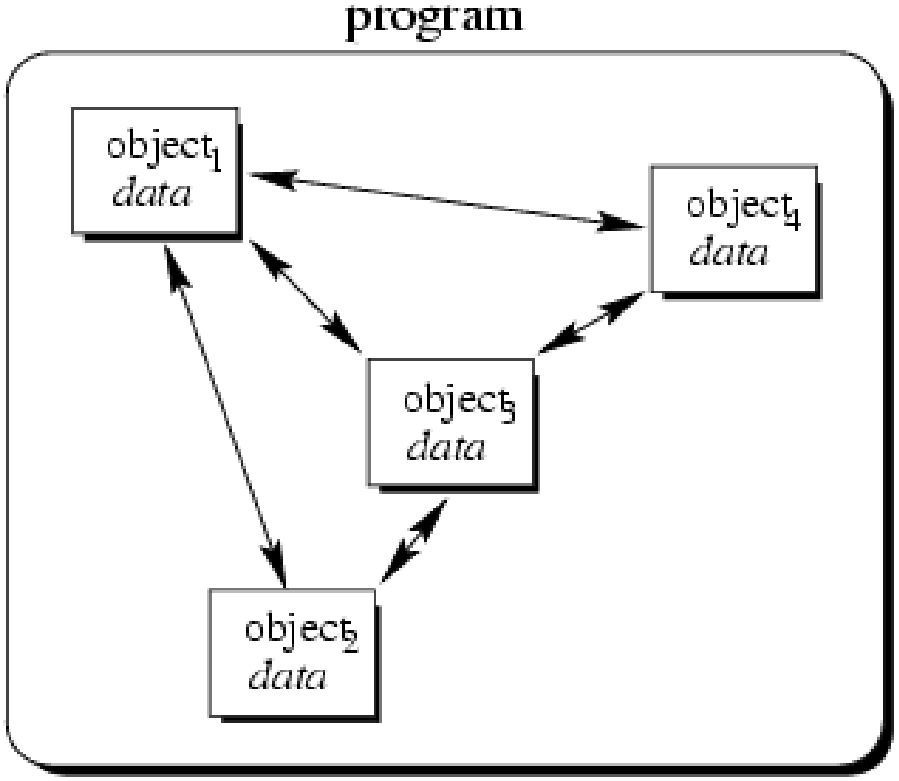
- O programa principal coordena as chamadas para os procedimentos e entrega os dados apropriados como parâmetros.

Linguagens Estruturadas (II)

- Linguagens Estruturadas
  - C, Pascal, Basic, Fortran
  - Facilidades para
    - Passar argumentos para funções
    - Retornar valores de funções
- Para o problema area de um retângulo, desenvolvemos uma função

```
int calcula_area (int l, int c) {
        retorno (l * c);
}
```

Linguagens Orientadas a Objetos (I)



- Objetos do programa interagem enviando mensagens (chamadas de métodos) uns aos outros

Linguagens Orientadas a Objetos (II)

Um objeto é um encapsulamento de funções e dados

- **Os objetos são abstrações**
  - Representam entidades do mundo real
  - As classes são os tipos de dados que definem as propriedades comuns compartilhadas entre vários objetos (atributos)
  - Objetos são instâncias de uma classe
- **Os objetos têm estado**
  - Tem um valor em um determinado momento

Linguagens Orientadas a Objetos (III)

Um objeto é um encapsulamento de funções e dados

- **Os objetos têm Operações**
  - Conjunto de operações chamados métodos associados que descrevem como realizar operações **(que podem ser chamados por outros objetos)**
- **Os objetos "enviam" Mensagens (chamadas de métodos em outros objetos)**
  - Solicitar um objeto para realizar uma de suas operações, enviando-lhe uma mensagem
  - Mensagens são os meios pelos quais os objetos solicitam que funcionalidades sejam realizadas e para trocar dados

Perspectiva OO

Perspectiva orientada a objetos:

- Definir um novo tipo Retângulo (a classe)
  - Dados
    - largura, comprimento
  - Função
    - area()
- Criar uma instância da classe (um objeto)
- Solicitar ao objeto a sua área

Em C++, ao invés de escrever um procedimento, definimos uma classe que encapsula o conhecimento necessário para responder à pergunta - aqui, o que é a área do retângulo.

Código exemplo orientado a objetos

```
class Retangulo {
  private:
   int largura, comprimento;
  public:
   Retangulo (int w, int l)
        {
         largura  = w;
         comprimento = l;
        }
   int area()  {
    return largura*comprimento;
   }
};
```

```
void main() {
   Retangulo ret(3, 5);
    cout << ret.area() << endl;
}
```

Linguagens Orientadas a Objetos

- Características de LPOO:
  - Encapsulamento
  - Herança
  - Polimorfismo
- LPOOs proporcionam:
  - Programação modular
  - Facilidade de Desenvolvimento
  - Manutenibilidade

Características de LPOOs

- **Encapsulamento:** Combina estrutura de dados com ações
  - Estrutura de dados: representa as propriedades, o estado, ou as características de objetos
  - Ações: comportamentos admissíveis que são controlados através dos membros das funções

Dados privados: Processo de fazer certos dados inacessíveis

Características de LPOOs

- **Herança:** Habilidade para derivar novos objetos a partir de outro
  - Permite que objetos de uma classe mais específica herdem os atributos (dados) e comportamentos (métodos) de uma classe mais geral / base
  - Capacidade para definir uma relação hierárquica entre objetos

Características de LPOO

- **Polimorfismo:** Habilidade que é dada a objetos diferentes de implementar operações de forma diferente dependendo do contexto de cada um porém atendendo ao contrato assumido ao implementar uma interface qualquer (do ponto de vista funcional)

C ++ básico

- Herda todas as características as ANSI C
- Pertencem todas as funções C
- Você não tem que escrever a programação POO em C ++

C ++ básico

- cin e cout (e #include <iostream>)

```
cout << “Teste";
char nome [10];
cin >> nome;
cout << “O nome " << nome << “ e um bonito nome." << endl;
cout << endl; // imprimir uma linha em branco
```

- Declarar variáveis em quase qualquer lugar

```
 // declarar uma variável quando você precisar dela
for (int k = 1; k < 5; k ++) {
    cout << k;
      }
```

C ++ básico

```
#include <iostream>
using namespace std;
int main() {
  int Num1;
  int Num2;
  cout << "Lendo o primeiro numero...: ";
  cin >> Num1;
  cout << endl;
  cout << "Lendo o segundo numero....: ";
  cin >> Num2;
  cout << endl;
  int soma = Num1 + Num2;
  cout << "Soma: " << soma;
  return 0;
}
```

C ++ Básico

- const
  - Em C se usava o #define
    - Preprocessador - Nenhuma verificação de tipo.
    - #define n 5
  - Em C ++, o especificador const
    - Verificação de tipo é aplicada pelo compilador
    - `const int n = 5; // declara e inicializa`

C ++ Básico

- Passando argumentos por referência usando “&”

```
#include <iostream>
using namespace std;

void addUm(int &ref) {
    ref = ref + 1;
}

int main() {
    int valor = 5;

    cout << "valor = " << valor << endl;
    addUm(valor);
    cout << "valor = " << valor << endl;
    return 0;
}
```

C ++ Básico

- C ++ permite sobrecarga de funções
  - Em C ++, as funções podem utilizar os mesmos nomes, dentro do mesmo âmbito de aplicação, onde cada um pode ser distinguido pelo seu nome *e* assinatura
  - A assinatura especifica o número, o tipo, e o fim dos parâmetros expressos como uma lista separada por vírgulas de tipos de argumento

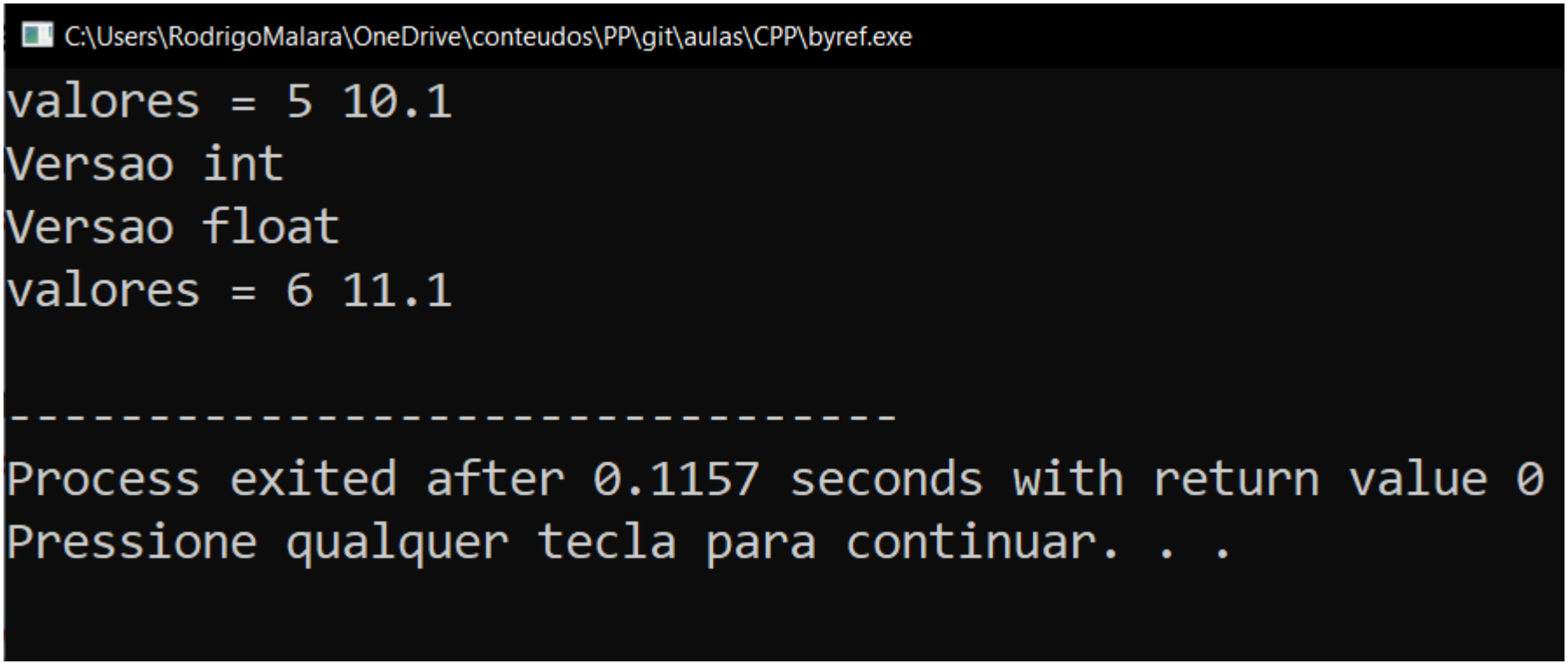
C ++ Básico

```
#include <iostream>
using namespace std;

void addUm(int &ref) {
    cout << "Versao int" << endl;
    ref = ref + 1;
}
void addUm(float &ref) {
    cout << "Versao float" << endl;
    ref = ref + 1;
}
int main() {
    int valor = 5;
    float valorF = 10.1;

    cout << "valores = " << valor << " " << valorF << endl;
    addUm(valor);
    addUm(valorF);
    cout << "valores = " << valor << " " << valorF << endl;
    return 0;
}
```

C ++ Básico


C:\Users\RodrigoMalara\OneDrive\conteudos\PP\git\aulas\CPP\byref.exe
valores = 5 10.1
Versao int
Versao float
valores = 6 11.1

Process exited after 0.1157 seconds with return value 0
Pressione qualquer tecla para continuar. . .

Exercício 1

- Escrever um programa que permita ao usuário digitar quatro notas bimestrais de um aluno, o número de aulas que o aluno assistiu e o número de aulas dadas no ano todo. Imprima se o aluno está aprovado, reprovado ou em recuperação, dadas as seguintes afirmações:
- Aluno aprovado: Aluno com média >= 7.0 e freqüência >=75%
- Aluno em recuperação: Aluno com média >=4.0 e média <7.0 e freqüência>=75%
- Aluno reprovado: Aluno com média< 4.0 ou freqüência <75%

Exercício 2

- Escrever uma função que calcule a área de um triangulo, dadas sua base e sua altura (int).
- Crie uma versão sobrecarregada dessa função que receba a base e a altura como 2 argumentos float.
- Faça um programa principal para testar as duas novas funções.