



# INF 1007 – Programação II

## Aula 01 - Módulos

Edirlei Soares de Lima  
<elima@inf.puc-rio.br>

# Módulos

- Um programa em C pode ser dividido em **vários arquivos fontes**
  - Programas reais precisam ser divididos em vários arquivos;
  - Seria intratável programar e manter o código de um programa grande em único arquivo;
- É possível criar um arquivo separado para implementar funções do programa. Esse arquivo é chamado de **módulo**;
- A implementação de um programa pode ser composta por um ou mais módulos;

# Módulos

- **Interface de um módulo de funções (Arquivo .h):**
  - Contem somente os protótipos das funções oferecidas pelo módulo;
  - Em geral possui o mesmo nome do módulo ao qual está associado;
  - Possui extensão .h
- **Módulo (Arquivo .c):**
  - Contem a implementação das funções que foram definidas na interface .h;
- **Programa Principal (Arquivo .c):**
  - Contem a função main do programa;
  - Inclui os módulos que foram criados e utiliza a suas funções;

# Módulos - Exemplo

- ***“Crie um programa para calcular a área e o volume de um cilindro”***
  - Podemos criar um módulo específico para implementar as funções relacionadas ao cilindro;
  - Dessa forma, o nosso programa será dividido em 3 arquivos:
    - **Geometria.h** – interface com a definição dos protótipos das funções `volume_cilindro` e `area_cilindro`;
    - **Geometria.c** – implementação das funções `volume_cilindro` e `area_cilindro` de acordo com o protótipo definido na interface;
    - **Principal.c** – implementação da função principal do programa;

# Módulos - Exemplo

- **Arquivo Geometria.h:**

```
#define PI 3.14159  
  
float volume_cilindro(float raio, float altura);  
  
float area_cilindro(float raio, float altura);
```

# Módulos - Exemplo

- **Arquivo Geometria.c:**

```
#include <math.h>
#include "Geometria.h"

float volume_cilindro(float raio, float altura)
{
    float volume = PI * pow(raio,2) * altura;
    return volume;
}

float area_cilindro(float raio, float altura)
{
    float area = 2 * PI * raio * (altura + raio);
    return area;
}
```

# Módulos - Exemplo

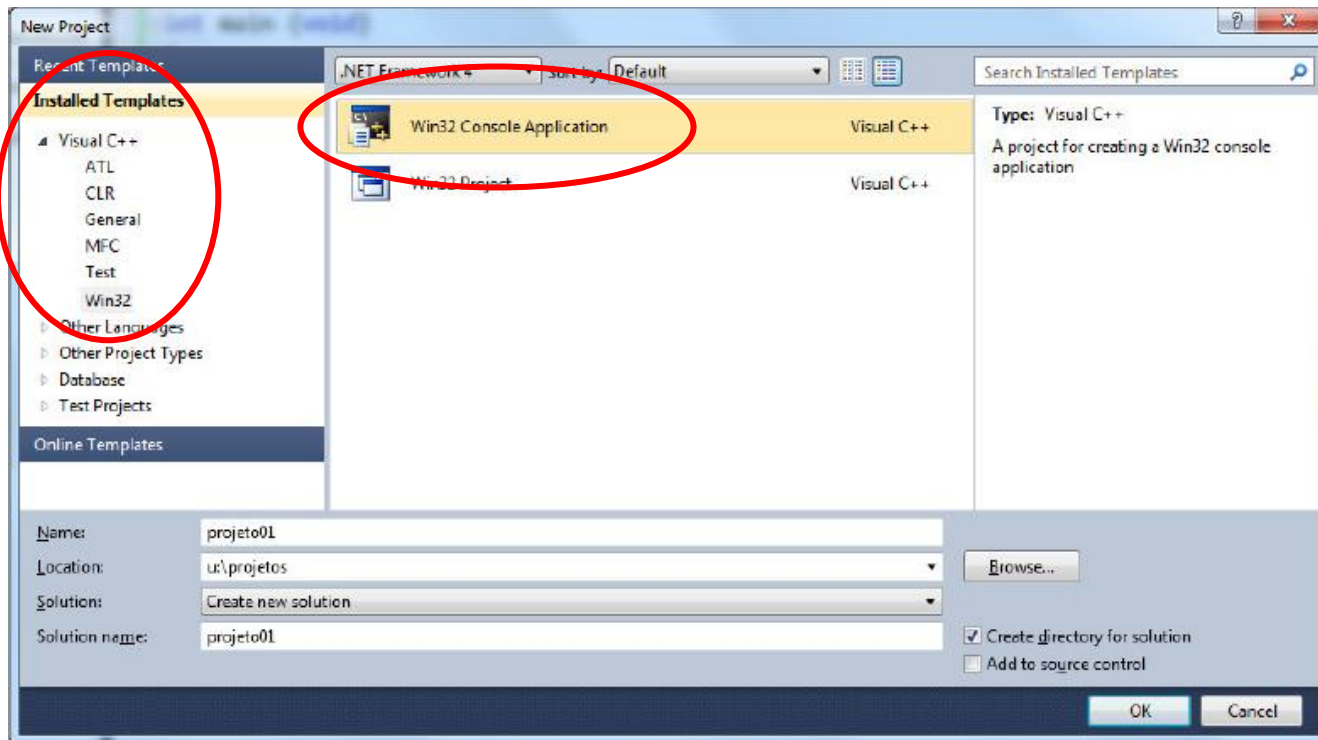
- **Arquivo Principal.c:**

```
#include <stdio.h>
#include "Geometria.h"

int main(void)
{
    float raio, altura, volume, area;
    printf("Entre com o valor do raio e da altura: ");
    scanf("%f %f", &raio, &altura);
    volume = volume_cilindro(raio, altura);
    area = area_cilindro(raio, altura);
    printf("Volume do cilindro: %f\n", volume);
    printf("Area do cilindro: %f\n", area);
    return 0;
}
```

# Criando Módulos no Visual Studio

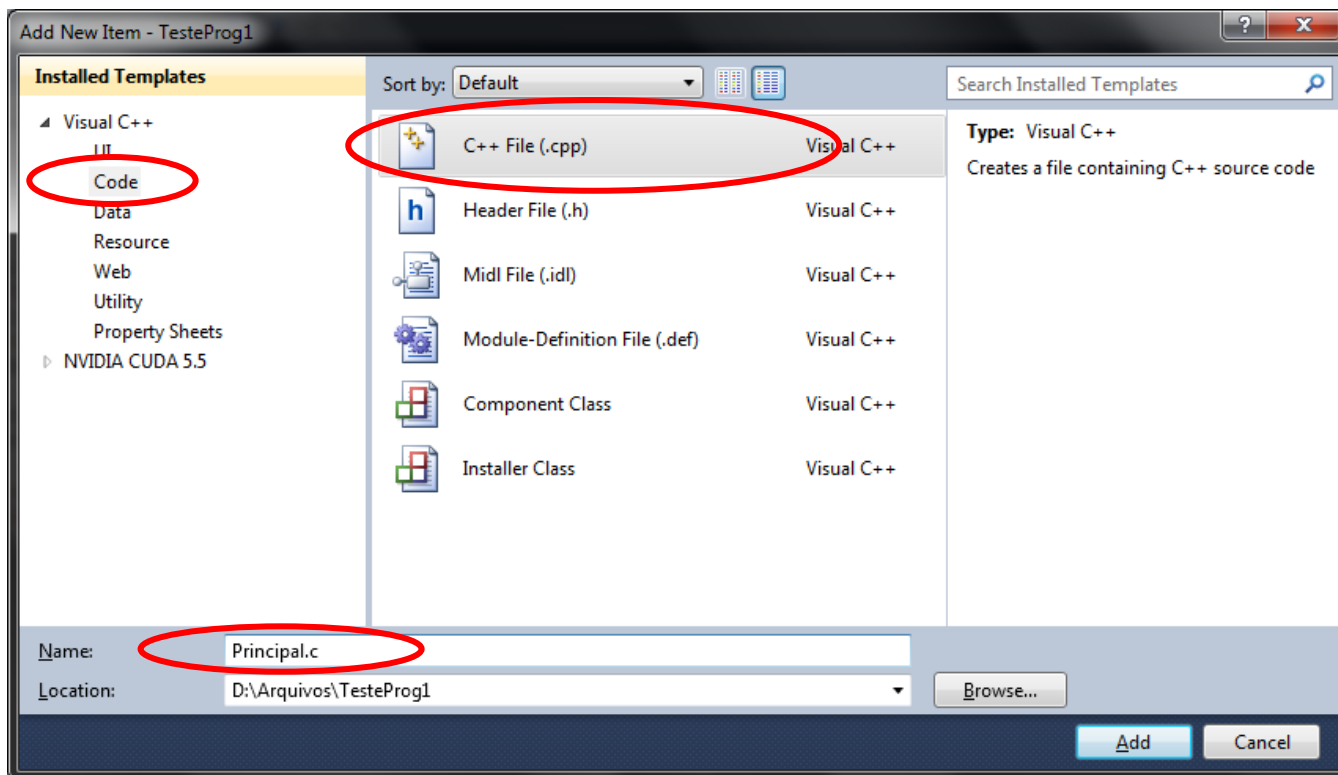
1. Crie um novo projeto no Visual Studio:





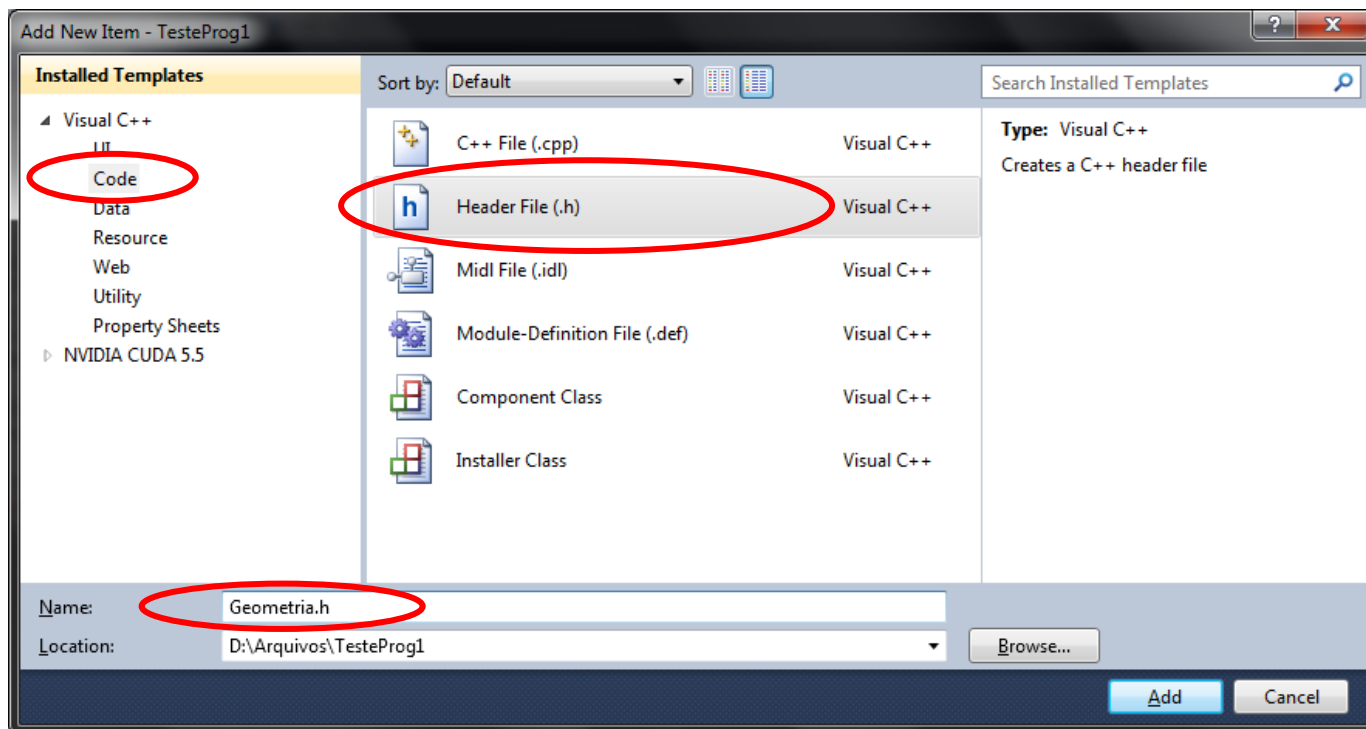
# Criando Módulos no Visual Studio

2. Adicione um novo arquivo fonte ao projeto:
  - Lembre-se de adicionar extensão .c no nome;
  - Esse será o arquivo que contém a função principal do programa;



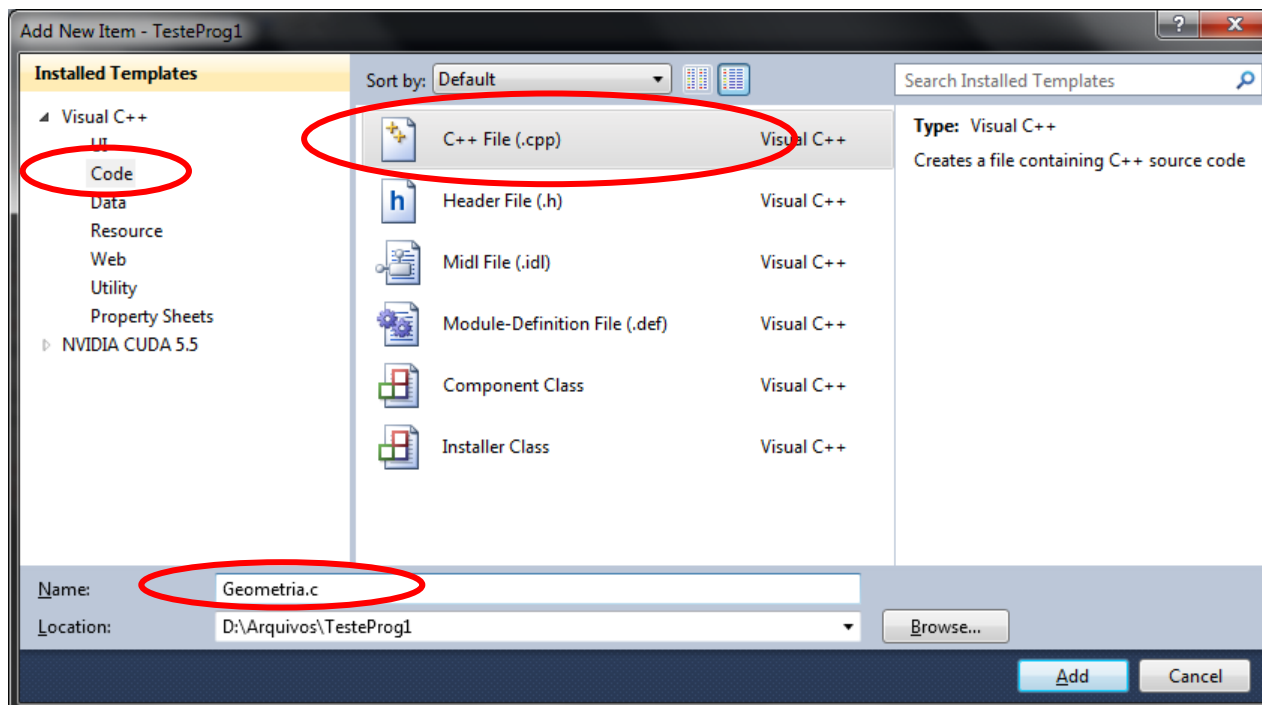
# Criando Módulos no Visual Studio

3. Adicione um novo arquivo do tipo **header** ao projeto:
  - Essa será a interface para o módulo que está sendo criado;

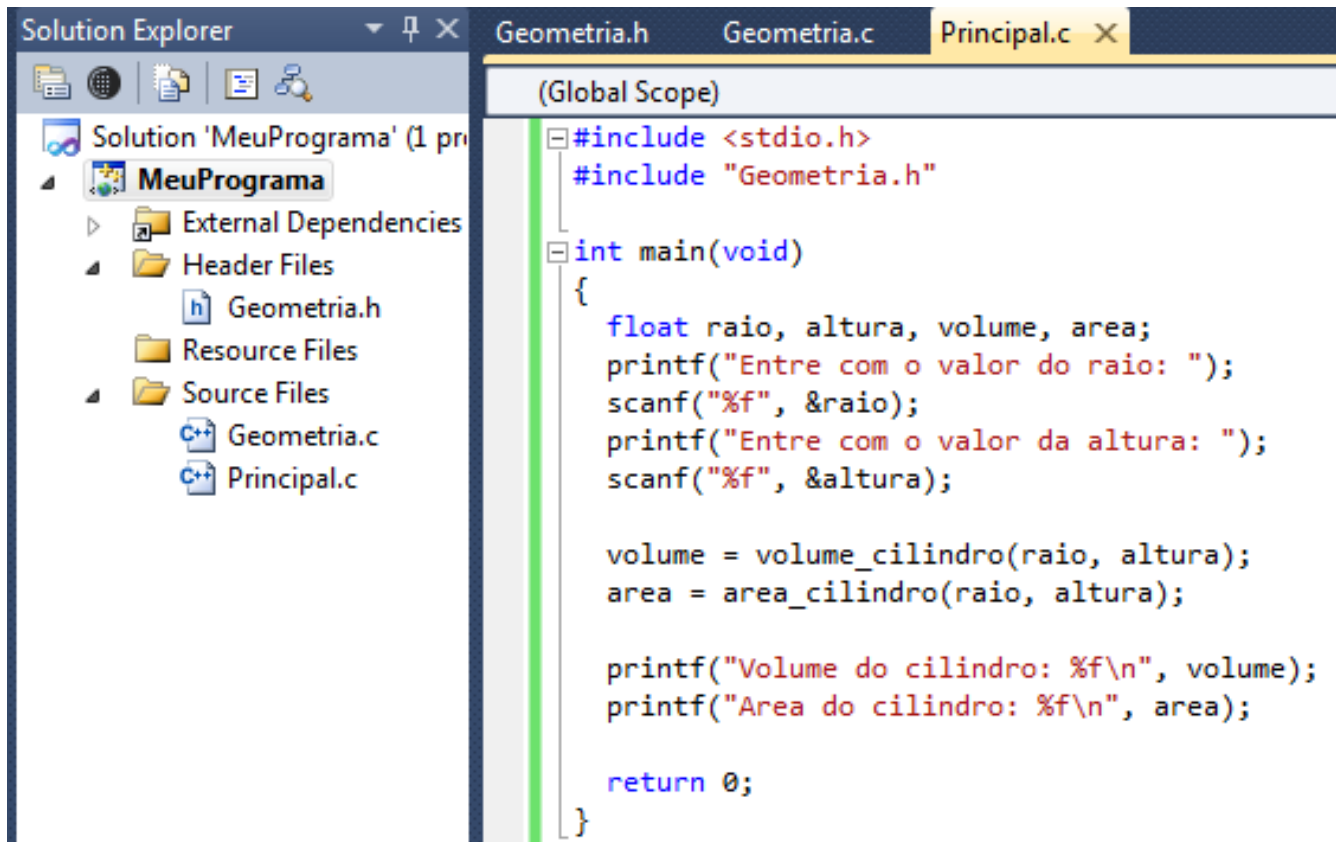


# Criando Módulos no Visual Studio

4. Adicione um novo arquivo fonte ao projeto :
- Essa será o arquivo que contem a implementação do módulo que está sendo criado;
  - Lembre-se de adicionar extensão .c no nome;



# Criando Módulos no Visual Studio



The image shows a screenshot of the Visual Studio IDE. On the left, the Solution Explorer displays a project named 'MeuPrograma' with the following structure:

- External Dependencies
- Header Files
  - Geometria.h
- Resource Files
- Source Files
  - Geometria.c
  - Principal.c

The main editor window shows the code for 'Principal.c' in the '(Global Scope)'. The code is as follows:

```
#include <stdio.h>
#include "Geometria.h"

int main(void)
{
    float raio, altura, volume, area;
    printf("Entre com o valor do raio: ");
    scanf("%f", &raio);
    printf("Entre com o valor da altura: ");
    scanf("%f", &altura);

    volume = volume_cilindro(raio, altura);
    area = area_cilindro(raio, altura);

    printf("Volume do cilindro: %f\n", volume);
    printf("Area do cilindro: %f\n", area);

    return 0;
}
```