

# Gabaritos (template)

## Funções template

- Gabaritos são funções especiais que podem operar com tipos genéricos e cuja funcionalidade pode ser adaptada a mais de um tipo ou classe sem repetir o código para cada tipo.
- Parâmetros `template`: tipo especial de parâmetro que pode ser usado para passar um tipo como argumento e não valores como os parâmetros regulares.
- Os parâmetros `template` são passados entre `<...>`, após o nome da função e antes dos parâmetros regulares entre `(...)`. Podem ser omitidos se puderem ser deduzidos a partir dos tipos dos parâmetros regulares.

```
template <class T>          cout << soma<int>(1, 3) << endl;
T soma(T P1, T P2)         cout << soma(1.3, 3.72) << endl;
{                           cout << soma<int>(1.3, 3.72) << endl;
    return P1+P2;          cout << soma(string("Adelardo"),
                           string(" Medeiros")) << endl;
}
```

## Classes template

- Gabaritos de classes permitem que uma classe possa ter membros que usem parâmetros `template` como tipos.
- Funções membro de classes `template` normalmente devem ser definidas como funções `template` para que possam usar parâmetros `template`.

```
template <class T>          template <class T>
class mypair {              T mypair<T>::getmax () {
    T a,b;                  return ( a>b ? a : b );
    public:                 }
    mypair (T first, T second):
        a(first),b(second){}
    T getmax();
};                           int main() {
                           mypair<int> X (100,75);
                           cout << X.getmax();
                           ...
}
```

## Compilação de gabaritos

- Gabaritos são compilados sob demanda, o que significa que o código de uma função `template` não é compilado até que seja necessária uma instanciação com argumentos `template` específicos. Nesse momento, quando uma instanciação é necessária, o compilador gera uma função especificamente para esses argumentos `template`.
- Como os `template` são compilados quando é necessário, a implementação (definição) de uma classe ou função `template` deve estar no mesmo arquivo que sua declaração: não se pode separar a implementação em um arquivo `.cpp` distinto do arquivo de cabeçalho `.h`.
- Os arquivos de inclusão `.h` para `template`'s usualmente contêm tanto a declaração quanto a implementação das funções gabarito.