

# Питањаца

- Колико ће класа компајлер направити на основу шаблона МојаКласа?

```
template<class T>  
class MojaKlasa { ... };
```

```
MojaKlasa<int> x;  
MojaKlasa<float> y;  
MojaKlasa<vector<int>> z;  
MojaKlasa<vector<long>> a;  
MojaKlasa<int> b;  
MojaKlasa<vector<logn>> c
```

# Питањаца

- Колико ће функција компајлер направити на основу шаблона `мојаФункција`?

```
template<class T>  
T мојаФункција(T x) { ... };
```

```
int Xint;  
float Xfloat;  
std::vector<int> Xvectorint;
```

```
мојаФункција(5)  
мојаФункција(Xint)  
мојаФункција(Xfloat)  
мојаФункција<double>(Xfloat)  
мојаФункција(Xvectorint)
```

# Питањаца

- Замислимо да нам се компајлер покварио и да не прихвата кључну реч `class`. Како да променимо доњи код, да заобиђемо тај необичан проблем са компајлером?

```
template<class T>
class MojaKlasa {
    public:
        MojaKlasa(T x) : m_x{x} {}
        void get() { return m_x*5; }
    private:
        T m_x;
};
```

# Питањаца

- Замислимо да нам се компајлер покварио и да прихвата да називи функција буду само `main` и `rega`, а не раде нам ни именски простори. Како да променимо доњи код, да заобиђемо тај проблем?

```
int foo(int x) { ... }
```

```
int bar(float x) { ... }
```

```
int main() {  
    int a;  
    float b;  
  
    ...  
  
    return foo(a) + bar(b);  
}
```

# Питањаца

- Замислимо да нам се компајлер покварио и да прихвата да називи функција буду само `main` и `rega`, а не раде нам ни именски простори. Како да променимо доњи код, да заобиђемо тај проблем?

```
int foo(int x) { ... }
```

```
int bar(int x) { ... }
```

```
int main() {  
    int a;  
    int b;  
  
    ...  
  
    return foo(a) + bar(b);  
}
```