

přetěžování funkcí (no)

co se stane v jazyce C při překladu následujícího kódu?

```
int ObvodCtverce(int aStrana)
{
    int Vysledek;
    Vysledek = 4 * aStrana;
    return Vysledek;
}
```

```
double ObvodCtverce(double aStrana)
{
    double Vysledek;
    Vysledek = 4 * aStrana;
    return Vysledek;
}
```

přetěžování funkcí (no)

- v C může být jen jediná funkce s daným jménem
- v C++ může být více stejnojmenných funkcí – přetěžování (viz. minulý příklad)
- přetěžování není myšleno ve smyslu překrytí (znemožnění přístupu k původní, ale přidání další funkce se stejným jménem na stejné úrovni)

```
int f(int);           // první z funkcí  
int f(int, int);      // přetížení – stejné jméno,  
// jiný počet parametrů
```

```
// použít lze obě  
a = f(b);  
c = f(d,e);
```

přetěžování funkcí (no)

- při volání vybírá překladač z přetížených funkcí/metod na základě kontextu (překladač si funkce interně „pojmenuje“ tak, že k názvu přidá typ parametrů např. @fYYXX, takže funkce pro něj potom nemají stejný název)
- funkce musí být odlišené: počtem nebo typem parametrů, prostorem,
- typ návratové hodnoty se nerozlišuje (nelze rozlišit pouze návratovou hodnotou)

```
int f(int);      // první z přetížených funkcí  
float f(int);   // nelze - návratová hodnota neodlišuje  
float f(float); // lze rozlišit - jiný typ parametru  
float f(float, int) // lze rozlišit - jiný počet parametrů
```

přetěžování funkcí (no)

- při vyhledávání má přednost "nejbližší", jinak musíme uvést celý přístup -
Prostor::Jméno
- problém s konverzemi

```
int f(int);    // první z přetížených funkcí
float f(int); // nelze - návratová hodnota neodlišuje
float f(float); // lze rozlišit - jiný typ parametru
float f(float, int) // lze rozlišit - jiný počet parametrů
```

```
float ff = f(3); // volání f(int)
```

```
f(3.14); // chyba - parametr double lze převést na int i float
        // a překladač nemůže jednoznačně rozhodnout
```

```
f( (float) 3.14); // s konverzí v pořádku - volá se f(float)
```

```
f(3,4.5); // OK, implicitní konverze parametrů
// volá se f(float, int) - rozlišení počtem parametrů
```