Deklaracija i definicija funkcije

- Funkcija može, a ne mora imati parametre. Funkcija koja nema parametre deklariše se kao *f()* ili *f(void)*, svejedno (radi se o istoj stvari)
- U tip funkcije ulazi povratni tip, kao i tipovi svih parametara, pri čemu se ne pravi razlika između sledećih tipova parametara:
  - niza elemenata tipa *T* sa dimenzijom ili bez nje i pokazivača na *T*
  - funkcije nekog tipa i pokazivača na funkciju istog tipa
  - parametra sa cv-kvalifikacijom i bez nje

Na primer, sledeće deklaracije istih imena su deklaracije istih funkcija (a deklaracije različitih imena su deklaracije različitih funkcija):

```
void f(int);
void f(const int);

void g(const int*);
void g(const int* const);

void h(int[]);
void h(int[5]);
void h(int*);
```

Deklaracija i definicija funkcije

- Povratni tip funkcije ne može biti funkcija ili niz (ali može biti pokazivač ili referenca na funkciju ili niz)
- Povratni tip funkcije može da se navede i iza parametara i znaka ->, što olakšava pisanje i čitanje deklaracija ako je povratni tip složen, ili ako se ne može odrediti, npr. zato što zavisi od tipova argumenata unutar šablona:

```
auto redirect (int(*)(int)) -> int(*)(int(*)(int));
```

```
template <typename U, typename V>
auto combine (U u, V v) -> decltype(u+v);
```

- Povratni tip ne mora da se navodi eksplicitno, nego da se ostavi prevodiocu da ga sam izvede, na osnovu tipa izraza iza naredbe *return*; svi tipovi iza višestrukih naredbi *return* moraju biti konzistentni; ova mogućnost nije dozvoljena za virtuelne funkcije:

```
template <typename U, typename V>
auto combine (U u, V v, double scalar1, double scalar2) {
  return u*scalar1 + v*scalar2;
}
```