Deklaracije

- Ista stvar može se iskoristiti za lakše pisanje koda bez obaveznog određivanja tipa rezultata izraza, koji može zavisiti od tipova operanada; na primer:

```
auto z = x*exp(y);
auto u = static_cast<decltype(v)>(z);
```

Tip varijable *z* zavisi od tipa varijable *x* i povratnog tipa one funkcije *exp* koja prima tip varijable *y*

- U nekim situacijama, posebno kod šablonskih funkcija, povratni tip funkcije je teško ili nemoguće odrediti. Tada se povratni tip funkcije može odrediti na osnovu tipa izraza i specifikatora *decltype* navedenog iza znaka ->:

```
template<typename T, typename U>
auto add(T t, U u) -> decltype(t + u) {
    return t+u;
}
```

Povratni tip ove funkcije određuje sam prevodilac, kao tip izraza *t+u*, koji zavisi od povratnog tipa operatora + definisanog za operande tipova *T* i *U* koji su parametri šablona

- Funkcija može imati povratni tip koji nije eksplicitno naveden, već se takođe zaključuje, ali na osnovu tipa izraza iza naredbe *return*:

```
template<typename T, typename U>
auto add(T t, U u) {
    return t+u;
}
```

Opseg važenja

- *Oblast* ili *opseg važenja* (*scope*) je deo teksta programa, ponekad diskontinualan (razdvojen delovima koda koji ne pripadaju tom opsegu) u kom je neko ime važeće, odnosno u kom se ono može koristiti neposredno, bez posebnih kvalifikacija, odnosno bez upotrebe operatora :: ili nekog drugog operatora ispred tog imena, npr. *Clock::setTime* ili *pc->setTime*
- Pod *imenom* (*name*) se podrazumeva identifikator, puno ime operatorske funkcije (npr. *operator+* ili *operator new*), ime korisnički definisanog konverzionog operatora (npr. *operator bool*) ili ime šablona sa stvarnim argumentima (npr. *stack<int>*)
- U opsegu važenja nekog imena, prevodilac može da izvrši tzv. *nekvalifikovanu potragu* (*unqualified lookup*) za deklaracijom tog imena, određujući tako entitet na koji se to ime odnosi. Na primer:

```
class Clock {
  …
  void setTime (int hh, int mm, int ss);
  …
  int h, m, s;
};

void Clock::setTime (int hh, int mm, int ss) {
  h = (hh>=0 && hh<=23) ? hh : 0;
  m = (mm>=0 && mm<=59) ? mm : 0;
  s = (ss>=0 && ss<=59) ? ss : 0;
}
```