

# Preklapanje funkcija

- ❖ U poređenje tipova ulaze i cv-kvalifikacije, što znači da se razlikuju tipovi koji jesu ili nisu kvalifikovani kao konstantni, uključujući i objekte i njihove funkcije članice koje jesu ili nisu konstantne. Na primer:

```
Task& TaskQueue::at (int position = 0);  
  
inline const Task& TaskQueue::at (int position = 0) const;  
    return const_cast<TaskQueue* const>(this)->at(position);  
}
```

Sada se na sledećim mestima pozivaju odgovarajuće funkcije, u zavisnosti od konstantnosti objekta za koji se one pozivaju:

```
const TaskQueue* pcq = ...;  
TaskQueue* pq = ...;
```

```
const Task& ct1 = pcq->at();
```

Poziva se *const Task& TaskQueue::at (int = 0) const*

```
const Task& ct2 = pq->at();
```

Poziva se *Task& TaskQueue::at (int = 0)*

```
Task& ct3 = pq->at();
```

Poziva se *Task& TaskQueue::at (int = 0)*

```
Task& ct4 = pcq->at();
```

Greška u prevođenju: poziva se *const Task& TaskQueue::at (int = 0) const*

# Preklapanje funkcija

- ❖ Jasno je da će se u slučaju da se formalni parametri neke od preklopljenih funkcija po broju i tipovima u potpunosti slažu sa stvarnim argumentima pozvati baš ta funkcija, ali šta ako takva funkcija ne postoji? Šta ako postoji jedna ili više funkcija koje se ipak mogu pozvati, uz dozvoljene implicitne konverzije tipova stvarnih argumenata u tipove formalnih parametara?
- ❖ Prevodilac sprovodi vrlo složen postupak potrage za funkcijom koja odgovara pozivu na osnovu tipova argumenata i parametara deklariranih funkcija (ADL i *overload resolution*). Ishod ovog postupka može da bude trojak:
  - Nijedna funkcija ne odgovara pozivu, odnosno nijedna se ne može pozvati čak ni implicitnim konverzijama tipova argumenata u tipove parametara; ovakav poziv onda nije ispravan i prevodilac prijavljuje grešku
  - Postoji više kandidata - funkcija koje se mogu pozvati uz implicitne konverzije tipova argumenata u tipove parametara, s tim da postupak ne preferencira nijednu od njih (nijedna ne odgovara više nego neka od ostalih); i ovaj poziv nije ispravan i prevodilac prijavljuje grešku tipa “višeznačan poziv funkcije” (*ambiguous function call*)
  - Jedna od funkcija - kandidata bolje odgovara od ostalih, na osnovu kriterijuma preferenciranja funkcija (ukoliko se neka potpuno poklapa po broju i tipovima parametara sa tipovima argumenata, onda je ona sigurno ta, ali to nije jedini slučaj); jedino ovakav poziv je ispravan i za njega će prevodilac generisati kod za poziv odabrane funkcije
- ❖ Ponekad prevodilac može jednoznačno odrediti funkciju koja se poziva, iako to čoveku koji tumači program možda nije najjasnije, jer su implicitne konverzije komplikovane i skrivene. Takve situacije treba izbegavati i takve pozive, odnosno preklopljene funkcije drugačije rešiti:
  - eksplicitnom konverzijom stvarnih argumenata u tačne tipove formalnih parametara, kako bi bilo nedvosmisleno jasno koja funkcija treba da se pozove
  - drugačijim imenovanjem funkcija, kako bi se poziv razrešio prostim imenovanjem funkcije