

---

# Inicijalizacija reference

---

- ❖ Reference obe vrste (i na lvrrednosti i na dvrednosti) mogu se inicijalizovati na sve navedene načine (u svim slučajevima umesto jednog znaka `&` za referencu na lvrrednost, može stajati i dva znaka `&&` za referencu na dvrednost):

- Kada se imenovana referenca deklariše sa inicijalizatorom:

```
T& ref = expression;  
T& ref(expression);  
T& ref = {expression, expression, ...};  
T& ref{expression, expression, ...};
```

- Kada se argument prenosi u pozvanu funkciju, a parametar je referenca:

```
void f (T& t);  
  
f(expression);  
f({expression, expression, ...});
```

- Kada je povratna vrednost funkcije referenca:

```
T& f () {  
    ...  
    return expression;  
}
```

- Kada se nestatički podatak član inicijalizuje u listi inicijalizatora članova u konstruktoru klase:

```
X::X (...) : ref(expression) {...}
```

---

# Inicijalizacija reference

---

- ❖ Ako se za inicijalizaciju upotrebljava lista, onda se primenjuje inicijalizacija listom: pravi se privremeni objekat tipa  $T$  i taj objekat se inicijalizuje datom listom:

```
int (&ref)[] = {0, 1, 2};
```

- ❖ Ako je referenca na lvrednost, a izraza kojim se inicijalizuje je lvrednost tipa  $T$  ili tipa izvedenog iz  $T$ , referenca se vezuje za objekat koji identifikuje taj izraz, onosno na njegov podobjekat osnovne klase tipa  $T$ :

```
Base& ref = *new Derived;
```

- ❖ Ako je referenca na lvrednost, može se inicijalizovati izrazom koji je dvrednost samo ako je referenca na konstantu; tada se opet pravi (materijalizuje) privremeni objekat za koji se vezuje referenca:

```
const int& ref = a+b;
```

- ❖ Takav privremeni objekat nastavlja da živi dok je na njega vezana referenca, ali ne nakon izlaska iz funkcije u kojoj se to dešava
- ❖ Izraz kojim se inicijalizuje referenca može biti i tipa za koji postoji konverzija koja vraća odgovarajuću referencu