

---

# Reference

---

- ❖ Reference i pokazivači na objekte imaju mnoge sličnosti:
  - u najvećem broju slučajeva implementiraju se na isti način: imaju vrednost adrese objekta za koji su vezani
  - pristup do objekta i preko pokazivača i preko reference je posredan
  - polimorfizam (dinamičko vezivanje) se aktivira pri posrednom pristupu do objekta i preko pokazivača i preko reference
  - mnoga pravila, a naročito pravila konverzije, npr. pokazivača / reference na izvedenu klasu u pokazivač / referencu na dostupnu osnovnu klasu, važe i za pokazivače i za reference na isti način
- ❖ Međutim, postoje značajne razlike između pokazivača i reference:
  - pokazivač se može preusmeriti tako da ukazuje na drugi objekat; referenca je od trenutka svog nastanka, tj. od inicijalizacije, trajno vezana za isti objekat
  - pokazivač može da ne ukazuje ni na šta (ima vrednost *null*); referenca uvek, od početka do kraja svog životnog veka, upućuje na jedan (isti) objekat
  - pristup do objekta preko pokazivača vrši se preko operatora `*`; pristup do objekta preko reference je neposredan, tj. ne zahteva nikakav operator – sama upotreba reference u izrazu odnosi se na referencirani objekat

---

# Reference

---

- ❖ U principu, reference se upotrebljavaju onda kada se želi notacija prenosa argumenta po vrednosti (umesto da se prenosi pokazivač kao “eksplicitan” posrednik), ali je potrebno da:
  - se izbegne kopiranje stvarnog argumenta u formalni
  - se podrži supstitucija, tj. da stvarni argument može biti specijalizacija tipa formalnog parametra
  - argument bude polimorfan
- ❖ Druga situacija jeste ta kada se želi da funkcija vraća nešto što se može dalje menjati, tipično kod preklapanja operatora, kako bi se ponašali slično operatorima za ugrađene tipove; na primer:

```
vector<complex> v(100);
```

```
v[i]++;
```