

Nabrajanja (enumeracije)

- ❖ Postoji implicitna konverzija iz tipa enumeracije u celobrojni tip, koja spada u skup celobrojnih promocija. Vrednost tipa enumeracije konvertuje se u:
 - prvi od sledećih tipova koji može da prihvati ceo opseg vrednosti te enumeracije, ukoliko nije eksplicitno zadat potporni tip: *int*, *unsigned int*, *long*, *unsigned long*, *long long*, or *unsigned long long*
 - u potporni tip enumeracije, kao i u tip u koji se on može promovisati, ako je potporni tip eksplicitno zadat
- ❖ Međutim, takva konverzija najčešće nije adekvatna, jer meša tipove i suprotna je navedenim principima. Samo u posebnim slučajevima koje treba dobro lokalizovati i enkapsulirati, ovo ima smisla. Na primer:

```
enum Status { initiated, suspended, committed, canceled, failed };  
int i = suspended; // i gets the value 1  
Status s = canceled;  
...
```

Ovo nije dobar način programiranja!

```
if (s==4) ... // conversion Status->int evaluates to true if s==failed
```

Nabrajanja (enumeracije)

- ❖ Vrednost koja se dobija ovom konverzijom definisana je na sledeći način:
 - ona koja je eksplicitno zadata inicijalizatorom (koji mora biti konstantni izraz)
 - ako inicijalizator nije zadat, jednaka je vrednosti prethodnog enumeratora plus 1; vrednost prvog enumeratora je podrazumevano 0, ako nije eksplicitno zadat inicijalizator:

```
enum Status { initiated = 1, suspended, //suspended == 2  
             committed, canceled = 5,  
             failed = committed+4 }; // failed == 7
```

- ❖ Suprotna konverzija iz celobrojnog tipa u tip enumeracije može se raditi samo eksplicitno, ne i implicitno; ukoliko data celobrojna vrednost ne spada u skup vrednosti enumeratora, ponašanje je nedefinisano; ovakve konverzije svakako nisu preporučljive:

```
Status s = 3;
```

```
Status s = static_cast<Status>(3);
```