Korisnički definisane konverzije

- Ukoliko se želi zabraniti implicitna korisnički definisana konverzija definisana konstruktorom ili operatorom konverzije, odgovarajuća funkcija (konstruktor ili operator) označava se specifikatorom *explicit*; takav konstruktor nije više konverzioni konstruktor, jer ne definiše implicitnu konverziju; na primer:

```
class X {
public:

   explicit X (bool);
   explicit operator bool ();

};

X f (X x) {
   bool b = x;

   return false;
}
```

Greška u prevođenju: nije dozvoljena implicitna konverzija iz *X* u *bool*

Greška u prevođenju: nije dozvoljena implicitna konverzija iz *bool* u *X*

Korisnički definisane konverzije

- Standardne konverzije mogu da se rade implicitno, i to tranzitivno (u nizu), pa jedna korisnički definisana konverzija iz tipa *X* u ugrađeni tip može da znači i implicitnu konverziju u neki drugi ugrađeni tip; ako se ovo želi sprečiti, ta druga konverzija deklariše se posebno, čime ona nadjačava tranzitivne konverzije, ali se označi kao *obrisana* (*deleted*), specifikatorom *=delete* iza zagrada, pa je prevodilac neće dozvoliti. Ovo može da spreči neke nepredviđene upotrebe objekata tipa *X*, na primer kada se želi upotreba tih objekata kao Bulovih vrednosti (tzv. "problem sigurnog tipa *bool*", *safe bool problem*):

```
class Assertion {
public:

  operator bool ();
  operator int () = delete;

};

void f (Assertion x) {

  x << 1;

  if (x) …

}
```

- Slično se može postići i deklarisanjem operatora konverzije u tip *bool* kao *explicit*, jer naredba *if* i naredbe petlji dozvoljavaju korisnički definisanu konverziju koja je eksplicitna:

```
class Assertion {
public:

  explicit operator bool ();
};

void f (Assertion x) {

  x << 1;

  if (x) …

}
```