Opseg važenja

- Nekvalifikovana pretraga imena podrazumeva da se za dato ime pretražuje određena oblast važenja i, ako se u njoj nađe deklaracija tog imena, prevodilac datu upotrebu imena vezuje za deklarisani entitet; u suprotnom, eventualno pretražuje neku drugu, okružujuću oblast važenja i u njoj traži deklaraciju tog imena itd. Ukoliko takvu deklaraciju ne nađe, upotreba imena nije ispravna
- Ovakva pretraga najpre po tekućoj oblasti važenja, pa onda po okružujućoj, može da se shvati i kao činjenica da ime deklarisano u ugnežđenoj oblasti važenja *sakriva* isto ime u okružujućoj oblasti važenja. Na primer:

```
int x = 0;

void f () {
  int x = 1;

  x = 3;
}

struct Dummy {
  int x;
};
```

Globalno ime *x* ima opseg važenja odavde do kraja ove jedinice prevođenja (fajla)

Početak oblasti važenja bloka

Lokalno ime *x* ima oblast važenja odavde do kraja ovog bloka i sakriva globalno *x*

Odnosi se na lokalno ime *x* iz oblasti važenja ovog bloka, a ne na globalno *x*

Završetak oblasti važenja bloka

Ovo ime *x* ima oblast važenja ove strukture/klase

- Dakle, u oblasti važenja, ime se može koristiti neposredno, nekvalifikovano. Van oblasti se može koristiti ili samo na određene načine, ili nikako
- Pojam oblasti važenja je isključivo vezana za tekst programa i vreme prevođenja, dakle statički koncept, bez semantike koja se odnosi na vreme izvršanja

Opseg važenja

- Opseg važenja imena počinje od njegove tzv. *tačke deklarisanja* (*point of declaration*), koja se razlikuje za različite entitete, na primer:
  - za enumeracije, klase ili šablone, to je odmah iza mesta identifikatora:

```
class X {
  X (const X&);
  …
};
```

  - za varijable, to je odmah iza deklaratora, a pre inicijalizatora:

```
const int x = 1, *p = &x;
{
  int* x[x] = {x};
}
```