Podrazumevani argumenti

- Podrazumevani argumenti nisu deo tipa funkcije. Ako se deklaracije funkcije ponavljaju, ne smeju ponovo navoditi podrazumevani argument za isti parametar, čak i ako je identičan. Na mestu poziva funkcije, podrazumevani argumenti predstavljaju uniju svih do tada deklarisanih podrazumevanih argumenata, s tim da ne sme postojati parametar koji nema podrazumevani argument iza parametra koji ima podrazumevani argument:

```
void f(int p1, int p2 = 2,int p3);
void f(int p1 = 1, int p2, int p3);
f(3);
void f(int p1, int p2,int p3=3);
f(0);
```

Greška u prevođenju: ponovljena definicija podrazumevanog argumenta

Greška u prevođenju: ne može da bude *void f(int=0,int=0,int)*

Poziv: *f(0,2,3)*

- U principu, ovaj koncept predstavlja notacionu pogodnost: da ga nema, bilo bi potrebno pisati više varijanti funkcija sa različitim parametrima; na primer, umesto:

```
double log (double x, double base=10.0);
```

moralo bi da se piše:

```
double log (double x, double base);
double log (double x) { return log(x,10.0); }
```

Preklapanje funkcija

- Ponekad postoji potreba da se naprave potprogrami koje rade logički istu stvar, samo sa drugačijim brojem ili tipovima parametara. U tradicionalnim jezicima, za ovakve potprograme morala bi da se osmisle različita imena, jer prevodilac poziv potprograma vezuje sa pozvanim potprogramom samo na osnovu imena
- Na jeziku C++, različite funkcije mogu imati isto ime, ukoliko se dovoljno razlikuju po broju ili tipovima parametara; ovo se naziva *preklapanje* (ili *preopterećenje*) funkcija (*function overloading*)
- Prevodilac vezuje poziv funkcije za pozvanu funkciju ne samo na osnovu imena, nego i u zavisnosti od broja i tipova stvarnih argumenata koje uparuje sa tipovima formalnih parametara, pri čemu se pretražuju oblasti važenja u zavisnosti od toga kojim oblastima važenja pripadaju argumenti (tzv. *argument-dependent lookup*, ADL i *overload resolution*). Na primer:

```
double max (double, double);
const char* max (const char*, const char*);
…
const char* s = max("March", "January");
double d = max(3.6,5);
```