Reference

- Slično je za sve druge upotrebe reference, na primer:
  - kada je referenca parametar funkcije, ona se inicijalizuje u trenutku poziva funkcije stvarnim argumentom, što znači da se ta referenca vezuje za stvarni argument; pošto se svaka upotreba reference u toj funkciji odnosi na referencirani objekat, svaka operacija u kojoj referenca učestvuje odnosi se na stvarni argument, pa se tako i postiže semantika prenosa po referenci:

```
void inc (int& i) {
  i++;
}

int main () {
  int s = 0;

  inc(s);
}
```

Pošto je *i* referenca, operacija se odnosi na referencirani objekat, a to je stvarni argument za koji je referenca vezana u trenutku svoje inicijalizacije, odnosno poziva funkcije

U trenutku poziva funkcije *inc*, formalni argument *i* inicijalizuje se stvarnim argumentom *s*; kako je *i* referenca, vezuje se za stvarni argument u trenutku svoje inicijalizacije, odnosno poziva funkcije

  - povratna vrednost funkcije takođe može biti referenca; povratna vrednost funkcije je tako referenca koja postoji kao bezimena vrednost, na mestu poziva funkcije; ona se inicijalizuje u trenutku povratka iz funkcije rezultatom izraza iza naredbe *return*; na taj način, ona se vezuje za objekat na koji se odnosi taj izraz; na pimer:

```
int& inc (int& i) {
  i++;
  return i;
}

int main () {
  int s = 0;

  inc(inc(s));
}
```

Funkcija vraća referencu koja se inicijalizuje u trenutku povratka iz funkcije i vezuje za objekat određen izrazom iza narebe *return*; pošto je u tom izrazu referenca, odnosi se na referencirani objekat, a to je stvarni argument

Rezultat prvog poziva funkcije *inc* je referenca koja upućuje na *s*; za njega se vezuje i formalni argument spoljašnjeg poziva, pa *s* na kraju ima vrednost *2*

Reference

- Iako se referenca ne može razvezati od objekta nekom operacijom (takva operacija ne postoji), ona, potpuno isto kao i pokazivač, može postati *viseća* (*dangling reference*), ukoliko objekat za koji je referenca vezana prestane da živi dok referenca još uvek živi; tipično, ako funkcija vrati referencu na automatski objekat koji nestaje izlaskom iz funkcije
- Kao i za pokazivače, obraćanje objektu na koga upućuje viseća referenca ima nedefinisane posledice
- Na primer, ovo ne valja, iako nije greška u prevođenju (mada prevodilac može izdati upozorenje ako prepozna ovaj problem):

```
int& f () {
  int r=1;
  …
  return r;
}

int& g (int i) {
  …
  return i;
}
```