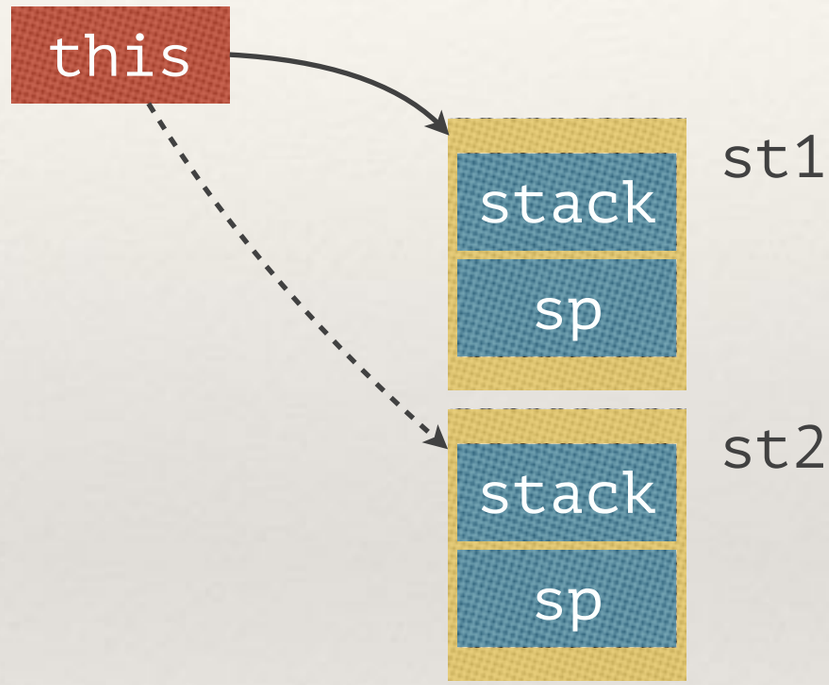
# Sa proceduralnog na OO programiranje: klase i objekti

- Sada se ovo koristi ovako:

```
#include "stack.h"

Stack* pSt1 = …;
stack_init(pSt1);
…
unsigned out;
…
stack_push(pSt1,in);
…
stack_pop(pSt1,&out);
…
Stack* pSt2 = …;
stack_init(pSt2);
…
stack_push(pSt2,in);
…
stack_pop(pSt2,&out);
…
```



# Sa proceduralnog na OO programiranje: klase i objekti

Upravo tako se na jeziku C++ implementiraju klase i objekti:

- Objekti se u vreme izvršavanja implementiraju kao instance obične C strukture koja sadrži samo vrednosti podataka članova date klase
- Za svaku (nestatičku) funkciju članicu klase, prevodilac generiše kod koji izgleda potpuno isto kao za "običnu" C funkciju, s tim što ta funkcija ima još jedan, prvi, skriveni argument *this* koji je pokazivač na ovu strukturu
- Svako neposredno obraćanje podatku članu unutar te funkcije članice:

```
sp = 0;
```

  prevodilac prevodi u implicitan pristup preko pokazivača *this*:

```
this->sp = 0;
```

- Svaki poziv funkcije članice za dati objekat:

```
pSt1->push(in);
```

  prevodilac prevodi u poziv "obične" C funkcije, pri čemu joj kao taj prvi argument *this* prenosi pokazivač na taj objekat:

```
stack_push(pSt1,in);
```