Object Oriented Design

Waarom objecten ?

- Functionaliteit en data op één plek
- Hergebruik
- Overerving van eigenschappen
- Uitbreiding
- Abstractie (verberg implementatie)
- Encapsulation (via private)
- Polymorfisme

Een recept voor cake

- 3 eieren
- 250 gram boter
- 250 gram bloem
- 250 gram suiker
- 1 zakje vanillesuiker
- snufje zout

Een recept voor cake

Gooi alles bij elkaar en maak er een mooi glad deeg van.

Giet het deeg in de cakevorm.

Zet de cakevorm in de oven en zet de oven aan op 160 graden.

Na een uur is de cake klaar.

Een recept voor cake

Kun je dit recept eten ?

Een recept voor cake

Kun je dit recept eten ?

Tja... als je het op eetbaar papier print...

Maar doe mij maar de cake !!!

Een recept voor cake

Kun je dit recept eten ?

Tja... als je het op eetbaar papier print...

Maar doe mij maar de cake !!!

Dus we zullen het recept moeten volgen en een cake bakken.

recept =====> cakes

class =====> objecten

```
class Cake
{
  int boter, bloem, eieren;

  int bak()
  {
    return boter+bloem+eieren;
  }

};
```

```
main()
{
Cake myCake;

  println( myCake.bak() );
}
```

Overerving : Inheritance

- Subclass voegt eigenschappen toe
- Bestaande eigenschappen worden overgenomen
- Sommige eigenschappen worden overschreven

Een speciaal geval

```
class ChocoladeCake extends Cake
{
  int cacao;

  int bak()
  {
    return boter+bloem+eieren+cacao;
  }

};
```

```
main()
{
Cake myCake;
ChocoladeCake specialCake;

println( myCake.bak() );
println( specialCake.bak() );

}
```