

*Skolan för Datavetenskap och kommunikation*

# PROGRAMMERINGSTEKNIK

## Föreläsning 10

# ARV

- Kap 11 i boken
- arv
- klassen object
- superklass
- subclass

*arv = inheritance*

# kodåtervinning

- Samma satser om igen?
  - *Skriv en slinga eller en funktion!*
- Samma funktioner i nytt program?
  - *Skriv en modul och importera den!*
- Samma data som skickas in i alla funktionerna?
  - *Skriv en klass!*
- Flera klasser med liknande attribut och metoder?
  - *Skriv en superklass och låt klasserna ärva från den!*

# Arv

```
class Subklass(Superklass):
```

Subklassen ärver alla

*attribut och metoder*

från klassen Superklass.

# object

När du skriver

```
class MinKlass(object):
```

betyder det att MinKlass ska ärvas från *object*

object är superklass till alla klasser i Python.

# Exempel: Bok

- *Bok* är den mest generella figuren - den får bli superklass
- *Ljudbok* är en sorts *Bok* - vi låter den vara subklass till *Bok*

# Klassen Bok

Attribut	Metoder
titel författare	__init__  __str__

```
class Bok:
```

```
    def __init__(self, titel, författare):  
        self.titel = titel  
        self.författare = författare
```

```
    def __str__(self):  
        return "'" + self.titel + "' skriven av " + self.författare
```





# Klassen Ljudbok

Attribut	Metoder
titel författare volym	__init__ __str__ sätt_volym

```
class Ljudbok(Bok):  
  
    def sätt_volym(self, v):  
        self.volym = v
```



```
def main():  
    bok1 = Bok("Kar", "Orhan Pamuk")  
    print(bok1)  
  
    bok2 = Ljudbok("Musicophilia", "O. Sacks")  
    print(bok2)  
  
    bok2.sätt_volym(5)  
    print(bok2.volym)
```

