

Overloading/overriding

Overloading: metoder med samme metodenavn, men forskjellig signatur

- Flere metoder kan ha samme navn så lenge antallet og/eller type parametere er annerledes

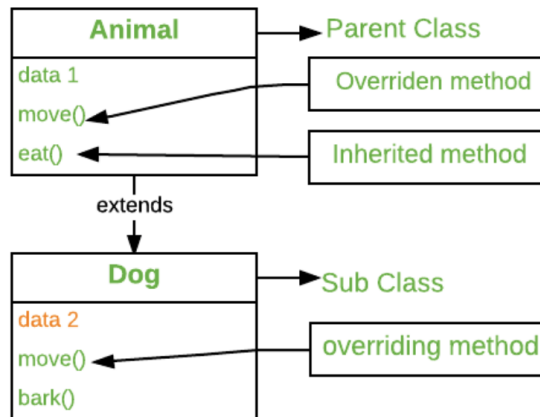
Metode	Signatur
public class settDimensjoner(int hoyde)	settDimensjoner(int)
public class settDimensjoner(int hoyde, int bredde)	settDimensjoner(int, int)

```
static int plusMethod(int x, int y) {
    return x + y;
}

static double plusMethod(double x, double y) {
    return x + y;
}

public static void main(String[] args) {
    int myNum1 = plusMethod(8, 5);
    double myNum2 = plusMethod(4.3, 6.26);
    System.out.println("int: " + myNum1);
    System.out.println("double: " + myNum2);
}
```

Override: versjonen av metoden som kjøres avhenger av objektet metoden blir kalt på



```
// Base Class
class Parent {
    void show( {
        System.out.println("Parent's show()");
    }
}

// Inherited class
class Child extends Parent {

    // This method overrides show() of Parent
    @Override
    void show() {
        System.out.println("Child's show()");
    }
}

// Driver class
class Main {
    public static void main(String[] args) {
        Parent obj1 = new Parent();
        obj1.show(); // "Parent's show()"

        Parent obj2 = new Child();
        obj2.show(); // "Child's show()"
    }
}
```

Noen regler:

- Hvis en metode ikke kan arves, kan den ikke overstyres.
- Like argumenter som metoden den *override* fra

- En konstruktør kan ikke *override*.
- **final** nøkkelord → kan ikke *override*
- Statiske metoder kan ikke brukes til å *override*.
- Returtypen må være den samme, eller en undertype av returtypen som er deklart i den opprinnelige overstyrte metoden i den overordnede klassen.