

toString(), equals()

Disse metodene kan man så selv redefinere til å gjøre noe mer fornuftig

toString()

→ Returnerer en String-representasjon av objektet

```
public static void main(String args[]) {
    Integer x = 5;
    System.out.println(x.toString()); //Her blir Integer gjort om til String
}
```

```
class Student{
    int id;
    String name;
    String city;

    //Her er det en metoden som returner objektet
    //i form av en String
    //overriding the toString() method
    public String toString(){
        return "Student: " +id+" "+name+" "+city;
    }

    public Student(int id, String name, String city){
        this.id=id;
        this.name=name;
        this.city=city;
    }
}
```

Equals()

→ sjekker om to objekter er like

- == sjekker om det peker på det samme (f.eks. Samme objekt)
- *.Equals* sjekker om det er samme "innhold"

```

public class Test {
    public static void main(String args[]) {
        String s1 = new String("HELLO");
        String s2 = new String("HELLO");
        System.out.println(s1 == s2);    // false
        System.out.println(s1.equals(s2)); // true
    }
}

```

```

class Punkt {
    protected int x, y;
    Punkt(int x0, int y0) {
        x = x0;
        y = y0;
    }
}

```

Anta:

```

Punkt p1 = new Punkt(3,4);
Punkt p2 = new Punkt(3,4);

Punkt2 q1 = new Punkt2(3,4);
Punkt2 q2 = new Punkt2(3,4);

```

Hva blir nå:

```

p1.toString(); Punkt@f5da06
p1.equals(p2); false

```

```

class Punkt2 {
    protected int x, y;
    Punkt2(int x0, int y0) {
        x = x0; y = y0;
    }

    public String toString() {
        return "x = "+x+" y = "+y;
    }

    public boolean equals(Object o) {
        if (!(o instanceof Punkt2))
            return false;
        Punkt2 p = (Punkt2) o;
        return x == p.x && y == p.y;
    }
}

```

```

q1.toString(); x=3 y=4
q1.equals(q2); true

```

35