Scanner

```
package no.hiof.fredrjen.oblig1;
import no.hiof.fredrjen.oblig1.models.Planet;
import java.util.Scanner;
import java.util.ArrayList;
public class Bonus3_1 {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.println("What is the name of the planet?: ");
      String name = scanner.nextLine();

      System.out.println("What is the radius of the planet (in km)?: ");
      int radius = scanner.nextInt();

      System.out.println("What is the mass of the planet (in kg)?: ");
      double mass = scanner.nextDouble();

      System.out.println();

      Planet planet = new Planet(name, radius, mass);
      planet.printDescription();
   }
}
```

konstruktør osv

```
public class Planet {
   private String name;
public String getName() {
      return name;
   }
 public void setName(String name) {
      this.name = name;
   }
  @Override
   public String toString() {
      return "skrift";
   }

        tilsvarer det som skal returneres
public double radiusInKm() {
return radius * 71492;
}
```

opphøyning

```
Math.pow(radiusInKm()*1000, 2);
```

Simple data types:

Byte: 8, -128.
short: 16 -32,768
int: 32 -2147,483,648...
long: 64 -9,223,372..
float: 32 -3,4e-0.38
double: 64 -1,7e-308..
char: 16 - complete charset
Boolean: true/false
variabel = datatype id = val;

Klasse:
public|final|abstract classname {class declarations
public static void main(String[] args){kode}
public method(){metoder}
}
this. - super

---

methods:
public|private - static. type|void name(argumenter){kode}

Compile og Run
javac fileName.java
java nameOfFile
Filnavn må tilsvare klassenavn nøyaktig. Konsensus er stor forbokstav på klasser.
ellers brukes liten forbokstav først og ord starter med stor forbokstav deretter.

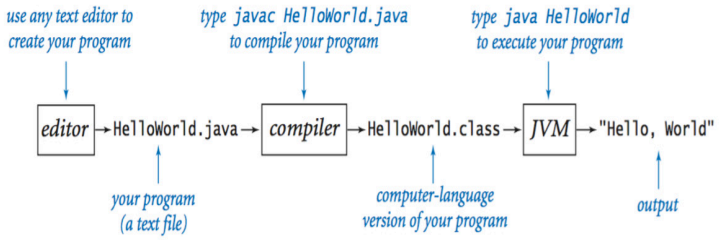
for-loop

```
public class Main {
   public static void main(String[] args) {
      Person person1 = new Person("Mike", 22, 1944);
       Person person2 = new Person("John", 44, 69420);
       ArrayList<Person> persons = new ArrayList<>();
       persons.add(person1);
       persons.add(person2);
       for (Person person : persons){
          System.out.println(person);
       }}
```

math.pow + konstruktør

```
public class Planet extends NaturalSatellite{
   public static final double JUPITER_RADIUS_IN_KM = 71492;
   public static final double JUPITER_MASS_IN_KG = 1.898E27;
   public static final double EARTH_RADIUS_IN_KM = 6371;
   public static final double EARTH_MASS_IN_KG = 5.972E24;

   public Planet(String name, double radius, double mass, double samiMajorAxis, double eccentricity,
            double orbitalPeriod, CelestialBody centralCelestialBody) {
      super(name, radius, mass, samiMajorAxis, eccentricity, orbitalPeriod, centralCelestialBody);
   }
   public double getSurfaceGravity() {
      // g = GM / R²
  return (CelestialBody.GRAVITATIONAL_CONSTANT * getMassInKg()) / Math.pow(getRadiusInMeter(), 2);
   }
   @Override
   public String toString() {
      return String.format("%s has a radius of %s Rjup and a mass of %s Mjup", getName(), getRadius(), get-
Mass());
  }}
```



En abstrakt klasse kan ikke instansieres. Vi kan ha både abstrakte og ikke-abstrakte metoder i en abstrakt klasse.

Abstrakt klasse

```
public abstract class CelestialBody {
   private String name;
   private double radius, mass;

   public static final double GRAVITATIONAL_CONSTANT = 6.67408E-11;

   public CelestialBody(String name, double radius, double mass) {
      this.name = name;
      this.radius = radius;
      this.mass = mass;
   }

   public abstract double getMassInKg();
   public abstract double getRadiusInKm();

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }
}
```

Abstrakt klasse som arver fra abstrakt klasse

```
public abstract class NaturalSatellite extends CelestialBody {
   private double semiMajorAxis, eccentricity, orbitalPeriod;
   private CelestialBody centralCelestialBody;

   public static final double ASTRONOMICAL_UNITS_IN_KM = 149597871;
   public static final double KM_TO_M = 1000;

   public NaturalSatellite(String name, double radius, double mass, double semiMa-
jorAxis, double eccentricity,
                double orbitalPeriod, CelestialBody centralCelestialBody) {
      super(name, radius, mass);
      this.semiMajorAxis = semiMajorAxis;
      this.eccentricity = eccentricity;
      this.orbitalPeriod = orbitalPeriod;
      this.centralCelestialBody = centralCelestialBody;
   }
```