

What is a class?

A class is a blueprint which describes a container for specific data.

```
public class BoringAnimal
{
}
```

- class definition
 - `public` keyword
 - `class` keyword
 - class name `BoringAnimal`
 - body of the class

Data in a class

The data contained within a class is represented by fields.

```
public class Dog
{
    public String name = "Griz";
    public int age = 5;
    public String breed = "Rottweiler";
}
```

- field definition
 - `public` keyword
 - type
 - name
 - value

Class Data Exercise

Write a class definition which represents your favorite movie. Include fields to contain:

- Name of the movie
- Year the movie was released
- Type of movie (Comedy, Action etc)

Make sure this class compiles with javac.

Class Data Exercise Example

```
public class FavoriteMovie
{
    public String name = "Deadpool";
    public int year = 2016;
    public String genre = "Comedy";
}
```

Interacting with the data

Logic contained in a class is called a method.

```
public class Pet
{
    public String animalType = "Dog";
    public String trick = "Roll Over";

    public void doTrick()
    {
        System.out.println(trick);
    }
}
```

Method Definition Breakdown

```
public void doTrick()  
{  
    System.out.println(trick);  
}
```

- method definition
 - `public` keyword
 - type signature
 - method body

Note: Type signatures can be much more complex than this, but that will be covered in a future class.

Method Exercise

Add a method to your movie class which will print each field on its own line.

Method Exercise Example

```
public class FavoriteMovie
{
    public String name = "Deadpool";
    public int year = 2016;
    public String genre = "Comedy";

    public void printFields()
    {
        System.out.println(name);
        System.out.println(year);
        System.out.println(genre);
    }
}
```


Objects

class: A blueprint for a data container.

object: A usable instance of the data container described by a class.
There can be more than one object created from a given class.

Understanding the distinction between a class and an object is critical to understanding java.

Class Instantiation

The process of creating an object from a class is called instantiation.

```
public class Pet
{
    public String animalType = "Dog";
    public String trick = "Roll Over";

    public void doTrick()
    {
        System.out.println(trick);
    }
}
```

Instantiation Breakdown

We can instantiate the class with the code:

```
Pet myPet = new Pet();
```

- instantiation
 - Type
 - name
 - new keyword
 - class name
 - parenthesis

Referencing Class Members

Field reference

```
System.out.println(myPet.animalType);
```

Invoking a method

```
myPet.doTrick()
```

Instantiation Exercise

Instantiate your movie class from the previous examples and invoke your method to print out the data.

Reference list

1. <https://docs.oracle.com/javase/tutorial/java/javaOO/index.html>
2. <http://web.mit.edu/1.00/www/definitions.htm>