

Objects in Java

Basic Concepts

- what is a class?
- defining classes
- creating objects

The Class definition is like a blueprint or design or specification



- here we specify the properties and actions of our class
- properties
 - name
 - breed
- actions
 - bark
 - wag tail

Create objects using class design

The tail shall be a discontinuous stream of feathers. It should be very thick at the base, gradually tapering toward the tip. All feathers barbs, and extending to the same gradually tapering toward the tip.

The tail shall be a discontinuous stream of feathers. It should be very thick at the base, gradually tapering toward the tip. All feathers barbs, and extending to the same gradually tapering toward the tip.

The ears shall be moderately sized for the head, set rather far back, and somewhat but not the shell slightly above eye level.

The back shall be strong and the spine to level from the withers to the croup, when standing or trotting.

The ribs shall be powerful and the spine to level from the withers to the croup, when standing or trotting.

The neck shall be of proper height to allow the dog to survey quite easily. It should be muscular.

The neck should rise straight from the shoulders with a moderate arch.

The forelimbs shall be straight, well developed and balanced with the hindquarters.

The hindquarters shall be straight, muscular and well developed. The hip of the hind limb well rounded and strong, stout hocks.

The hind feet shall be strongly hooved, rounded with moderate curvature at the tips, and powerful, clean, defined digits.

Feet are strong and compact, with well arched toes and a well-developed pad.

Skull: Moderately convex, tapering toward occiput, well defined and alertness on a back view of the head. There should be no patches on, set well apart, and neither protruding nor deep set.

The head shall be elegant and free from faulty showing. The base of the skull shall be straight, the eye set far from the center of the skull.

The eyes shall be wide and the pupils well developed.

The ears shall be powerful and free from substance that would interfere with their function and also not overly large.

The neck shall be of proper height to allow the dog to survey quite easily. It should be muscular.

The neck should rise straight from the shoulders with a moderate arch.

The forelimbs shall be straight, well developed and balanced with the hindquarters.

The hindquarters shall be straight, muscular and well developed. The hip of the hind limb well rounded and strong, stout hocks.

The hind feet shall be strongly hooved, rounded with moderate curvature at the tips, and powerful, clean, defined digits.

Feet are strong and compact, with well arched toes and a well-developed pad.

ELEVATION
NOT TO SCALE

PERSONAL INFORMATION

NAME: _____

SEX: _____

BORN: _____

HAIR: _____

LOCATION: _____

MOTHER: _____

FATHER: _____

FAV. HONEY: _____

FAV. JOG: _____

WEIGHT: _____

COLOR: _____

GENERAL NOTES

BREED: Labrador Retriever

HISTORY: The Labrador Retriever did not come from Labrador, but from Newfoundland. The man was popularized with small water dogs, when first used, they eventually produced a breed referred to as the St. John's Water Dog, a precursor for the Lab of today. The breed eventually found its way to Newfoundland from the British Isles and eventually to the United States. Many Labs have interbred with other types of retrievers that led to the breed we know today.

GENERAL APPEARANCE: The Labrador Retriever is a strong built, medium-sized, short-coated, dog possessing a sound, athletic, well-balanced conformation.

PROPORTION: When standing, length from the shoulder to the rump is equal to or slightly longer than the distance from the withers to the ground. Standing from the elbow to the ground should be equal to one-half the height at the withers.

SIZE: Bitches: 21 1/2 to 24 inches (54 to 61 inches) (1)
Males: 22 1/2 to 26 inches (57 to 66 inches) (2)
Weight: 45 to 60 lbs (20 to 27 kg) (1)

COAT: Short, straight and very dense. A soft, feathered ruffian and profuse protection from water and cold.

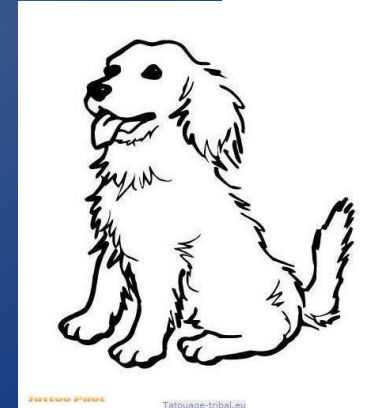
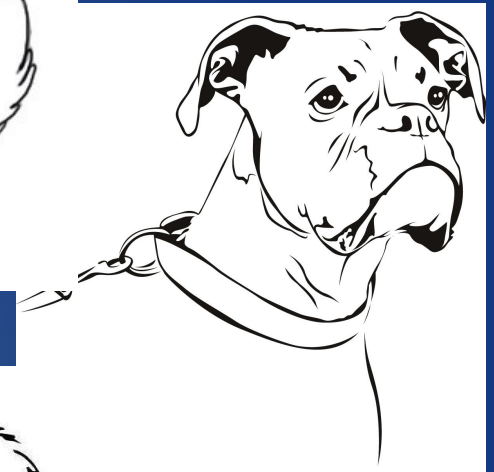
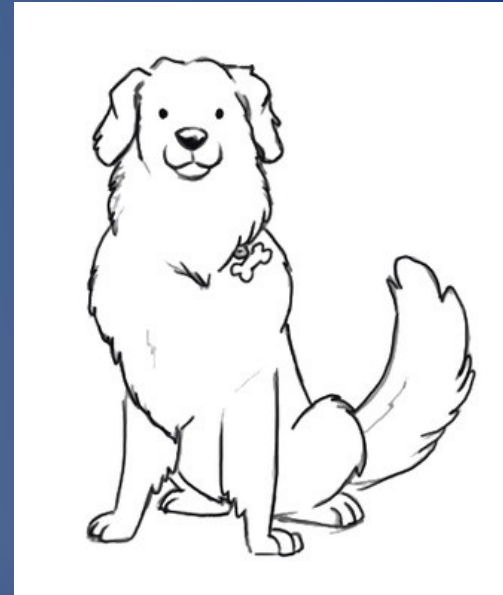
COLOR: Black, yellow or chocolate.

TEMPERAMENT: A kind, outgoing, free-willed, amiable, eager to please and non-aggressive towards man or animal. His gentle, merry, intelligent and adaptable makes him an ideal dog.

FRANK LLOYD WOOD
CANINE ARCHITECT

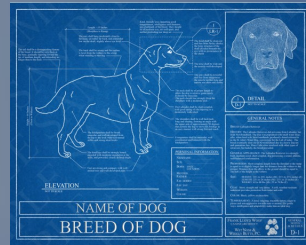
WET NOSE & WOOLY BUTTS, P.C.

GENERAL NOTES & DETAILS
D-1



Create objects using class design

File: Dog.java



```
class Dog
{
    String name;
    int age;
    String breed;
    int tailPosition = 0;

    void bark()
    {
        println("woof!");
    }

    void wagTail()
    {
        tailPosition = -5;
        delay(1); // 1 second
        tailPosition = +5;
        delay(1); // 1 second
        tailPosition = 0;
    }
}
```

File: DogPark.java

```
public static void main(...)
{
    // create some dog objects
    Dog dog1 =
        new Dog("Fido", "Lab");

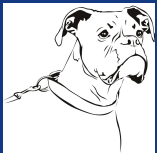
    Dog dog2 =
        new Dog("Rex", "Boxer");

    Dog dog3 =
        new Dog("Fido", "Spaniel");

    dog1.bark();

    dog2.wagTail();

    dog3.bark();
    dog3.wagTail();
}
```



Defining a Class

- define a class in a separate file with the same name as the class
 - Person.java
- define fields to hold data, or properties, of the class
 - name, age

```
class Person
{
    String name;
    int age;
}
```

Using a Class to Create Objects

- a class is just an idea
- an object is that idea made into something "real"
- create and use objects in a separate file from the class
 - TestPerson.java
- the "new" keyword asks for space in memory for object

```
// a regular old variable  
int count = 0;
```

```
// more complex variables  
String msg = "Hello";
```

```
double[] grades =  
    new double[4];
```

```
// create a new person  
Person p1 = new Person();
```


Object Data Fields

- data fields contain the properties of individual objects
- each object will have its own copies of its own data
- data fields can store basic data types, arrays, or even other objects

```
// create a new person
Person p1 = new Person();
Person p2 = new Person();

p1.name = "Arthur Dent";
p1.age = 44;

p2.name = "Ford Prefect";
p2.age = 32;

println(p1.name);
// output is "Arthur Dent"

println(p2.age);
// output is 32
```