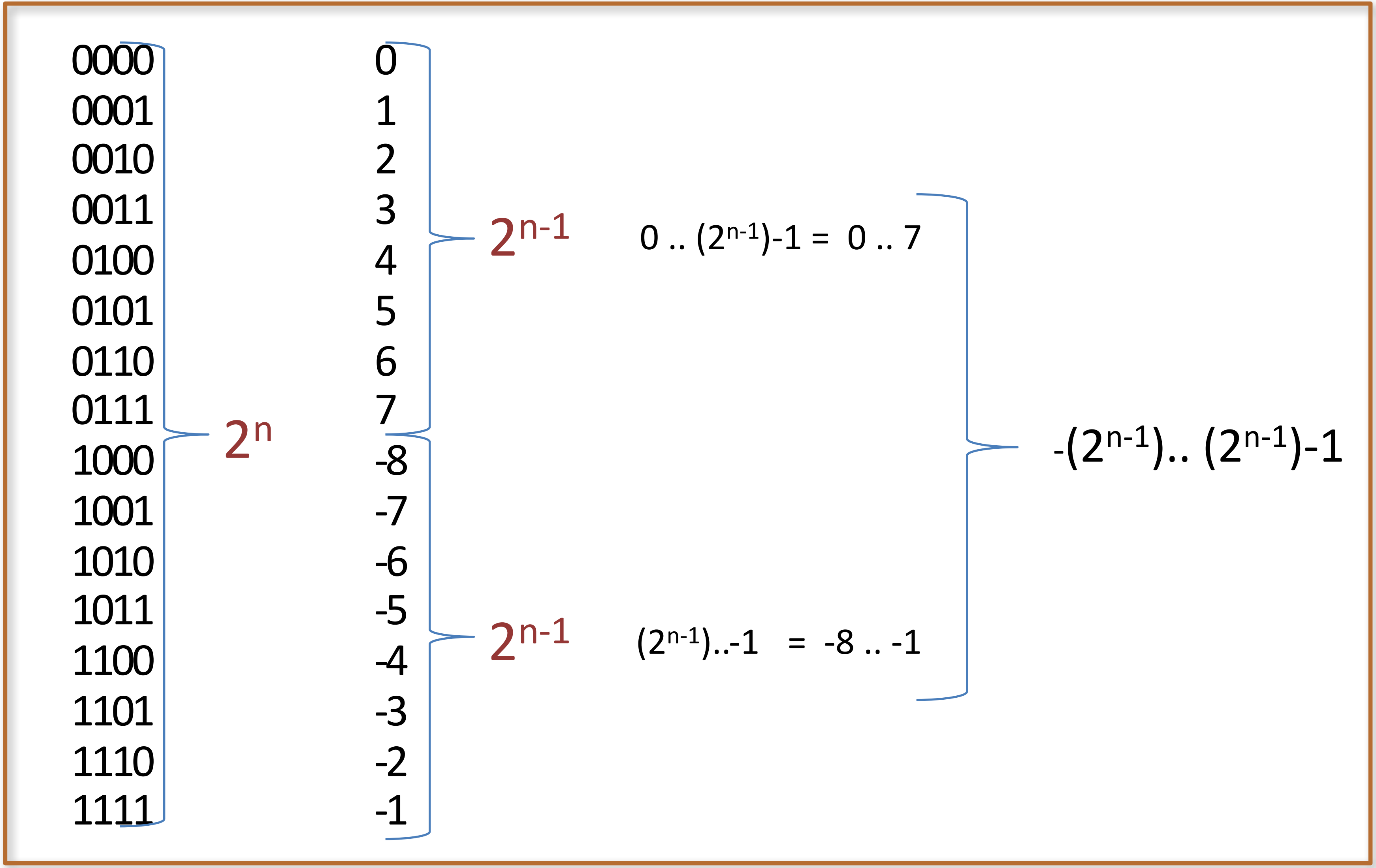
Classes and Objects

Primitives

Single-valued data items. Not Objects

byte	8-bit signed two's complement integer
short	16-bit signed two's complement integer
int	32-bit signed two's complement integer
long	64-bit signed two's complement integer
float	32-bit single-precision IEEE 754
double	64-bit single-precision IEEE 754
boolean	true/ false
char	16-bit Unicode character

The number of bits and the range of values
Two's complement integers, n=4



0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1

2^n

2^{n-1} $0 .. (2^{n-1})-1 = 0 .. 7$

2^{n-1} $(2^{n-1})..-1 = -8 .. -1$

$-(2^{n-1}).. (2^{n-1})-1$

The number of bits and the range of values
Two's complement integers, n=32

$-(2^{n-1}) .. (2^{n-1})-1$

$-(2^{32-1}) .. (2^{32-1})-1$

$-(2^{31}) .. (2^{31})-1$

-2,147,483,648 .. +2,147,483,647

-OR-

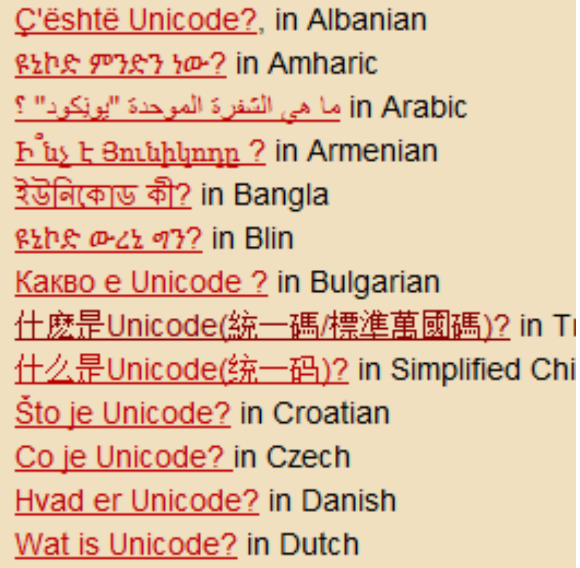
2^{10} = 1,024 or about 1,000 (Kilo)
2^{20} = $1{,}024^2$ or about 1,000,000 (Mega)
2^{30} = $1{,}024^3$ or about 1,000,000,000 (Giga)
2^{40} = $1{,}024^4$ or about 1,000,000,000,000 (Tera)

$2^{31} = 2^1 * 2^{30}$
= 2 G (about 2 billion)

So, an integer can represent +/- 2 billion

char

ASCII	8-bit code	$2^8 = 256$ chars
Unicode	16-bit code	$2^{16} = 2^6 * 2^{10} = 64{,}000$ chars



Ç'është Unicode?, in Albanian

ዩኒኮድ ምንድን ነው? in Amharic

ما هي الشفرة الموحدة "يونِكود" ؟ in Arabic

Ի՞նչ է Յունիկոդը ? in Armenian

ইউনিকোড কী? in Bangla

ዩኒኮድ ውረኒ ግን? in Blin

Какво е Unicode ? in Bulgarian

什麼是Unicode(統一碼/標準萬國碼)? in Tr

什么是Unicode(统一码)? in Simplified Chi

Što je Unicode? in Croatian

Co je Unicode? in Czech

Hvad er Unicode? in Danish

Wat is Unicode? in Dutch

http://www.unicode.org/charts/

	170	171
0	ᜀ 1700	ᜐ 1710
1	ᜁ 1701	ᜑ 1711
2	ᜂ 1702	◌ᜒ 1712
3	ᜃ 1703	◌ᜓ 1713
4	ᜄ 1704	◌᜔ 1714
5	ᜅ 1705	▨
6	ᜆ 1706	▨
7	ᜇ 1707	▨

Independent vowels

1700	ᜀ	TAGALOG LETTER A
1701	ᜁ	TAGALOG LETTER I
1702	ᜂ	TAGALOG LETTER U

Consonants

1703	ᜃ	TAGALOG LETTER KA
1704	ᜄ	TAGALOG LETTER GA
1705	ᜅ	TAGALOG LETTER NGA
1706	ᜆ	TAGALOG LETTER TA
1707	ᜇ	TAGALOG LETTER DA
1708	ᜈ	TAGALOG LETTER NA
1709	ᜉ	TAGALOG LETTER PA
170A	ᜊ	TAGALOG LETTER BA
170B	ᜋ	TAGALOG LETTER MA
170C	ᜌ	TAGALOG LETTER YA
170D	▨	<reserved>
170E	ᜎ	TAGALOG LETTER LA
170F	ᜏ	TAGALOG LETTER WA
1710	ᜐ	TAGALOG LETTER SA
1711	ᜑ	TAGALOG LETTER HA

Dependent vowel signs

1712	◌ᜒ	TAGALOG VOWEL SIGN I
1713	◌ᜓ	TAGALOG VOWEL SIGN U

Virama

1714	◌᜔	TAGALOG SIGN VIRAMA

	13A	13B	13C	13D	13E	13F
0	Ꭰ 13A0	Ꮀ 13B0	Ꮐ 13C0	Ꮠ 13D0	Ꮰ 13E0	Ᏸ 13F0
1	Ꭱ 13A1	Ꮁ 13B1	Ꮑ 13C1	Ꮡ 13D1	Ꮱ 13E1	Ᏹ 13F1
2	Ꭲ 13A2	Ꮂ 13B2	Ꮒ 13C2	Ꮢ 13D2	Ꮲ 13E2	Ᏺ 13F2
3	Ꭳ 13A3	Ꮃ 13B3	Ꮓ 13C3	Ꮣ 13D3	Ꮳ 13E3	Ᏻ 13F3
4	Ꭴ 13A4	Ꮄ 13B4	Ꮔ 13C4	Ꮤ 13D4	Ꮴ 13E4	Ᏼ 13F4
5	Ꭵ 13A5	Ꮅ 13B5	Ꮕ 13C5	Ꮥ 13D5	Ꮵ 13E5	▨
6	Ꭶ 13A6	Ꮆ 13B6	Ꮖ 13C6	Ꮦ 13D6	Ꮶ 13E6	▨
7	Ꭷ	Ꮇ	Ꮗ	Ꮧ	Ꮷ	▨

	0F0	0F1	0F2	0F3	0F4	0F5	0F6	0F7	0F8	0F9	0FA	0FB	0FC	0FD	0FE	0FF
0	ༀ 0F00	༐ 0F10	༠ 0F20	༰ 0F30	ཀ 0F40	ཐ 0F50	འ 0F60	▨	◌ྀ 0F80	◌ྐ 0F90	◌ྠ 0FA0	◌ྰ 0FB0	࿀ 0FC0	࿐ 0FD0	▨	▨
1	༁ 0F01	༑ 0F11	༡ 0F21	༱ 0F31	ཁ 0F41	ད 0F51	ཡ 0F61	◌ཱ 0F71	◌ཱྀ 0F81	◌ྑ 0F91	◌ྡ 0FA1	◌ྱ 0FB1	࿁ 0FC1	࿑ 0FD1	▨	▨
2	༂ 0F02	༒ 0F12	༢ 0F22	༲ 0F32	ག 0F42	དྷ 0F52	ར 0F62	◌ི 0F72	◌ྂ 0F82	◌ྒ 0F92	◌ྡྷ 0FA2	◌ྲ 0FB2	࿂ 0FC2	࿒ 0FD2	▨	▨
3	༃ 0F03	༓ 0F13	༣ 0F23	༳ 0F33	གྷ 0F43	ན 0F53	ལ 0F63	◌ཱི 0F73	◌ྃ 0F83	◌ྒྷ 0F93	◌ྣ 0FA3	◌ླ 0FB3	࿃ 0FC3	࿓ 0FD3	▨	▨
4	༄ 0F04	༔ 0F14	༤ 0F24	༴ 0F34	ང 0F44	པ 0F54	ཤ 0F64	◌ུ 0F74	◌྄ 0F84	◌ྔ 0F94	◌ྤ 0FA4	◌ྴ 0FB4	࿄ 0FC4	࿔ 0FD4	▨	▨
5	༅ 0F05	༕ 0F15	༥ 0F25	◌༵ 0F35	ཅ 0F45	ཕ 0F55	ཥ 0F65	◌ཱུ 0F75	྅ 0F85	◌ྕ 0F95	◌ྥ 0FA5	◌ྵ 0FB5	࿅ 0FC5	࿕ 0FD5	▨	▨
6	༆ 0F06	༖ 0F16	༦ 0F26	༶ 0F36	ཆ 0F46	བ 0F56	ས 0F66	◌ྲྀ 0F76	◌྆ 0F86	◌ྖ 0F96	◌ྦ 0FA6	◌ྶ 0FB6	◌࿆ 0FC6	࿖ 0FD6	▨	▨
7	༇ 0F07	༗ 0F17	༧ 0F27	◌༷ 0F37	ཇ 0F47	བྷ 0F57	ཧ 0F67	◌ཷ 0F77	◌྇ 0F87	◌ྗ 0F97	◌ྦྷ 0FA7	◌ྷ 0FB7	࿇ 0FC7	࿗ 0FD7	▨	▨
8	༈	◌༘	༨	༸	▨	མ	ཨ	◌ླྀ	ྈ	▨	◌ྨ	◌ྸ	࿈	࿘	▨	▨

Class

a blueprint or template that describes the properties and behaviors of an object.

Object

an instance of a class which has specific properties

Example:

class Cat

Properites: breed, gender, color, number of legs, neutered,…

Behaviors: meows, eats, purrs, sleeps, …

A specific cat is an *instance* of the Cat class:

Breed: tabby
gender: male
Color: gray
Number of legs: 4
Neutered: yes

Breed: calico
gender: female
Color: brown
Number of legs: 4
Neutered: yes

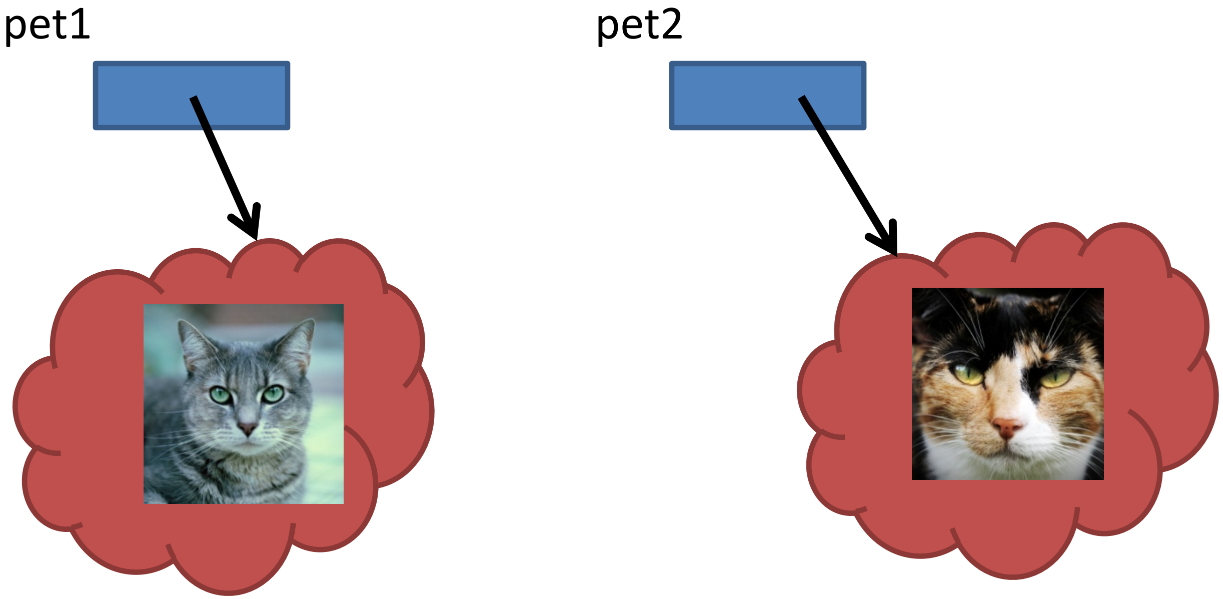
Breed: Maine Coon
gender: male
Color: gray
Number of legs: 4
Neutered: yes

Breed: Siamese
gender: female
Color: sable
Number of legs: 4
Neutered: yes

Instantiation- making an *instance* of a class

```
Cat pet1, pet2, pet3;


pet1 = new Cat("tabby","male","gray",4,true);
pet2 = new Cat("calico","female","brown",4,true);
```



pet1

pet2

Static vs. Instance

Static variables belong to a *class.*

There is only ONE variable for EVERY instance of the class.

Instance variables belong to an object

The object is an instance *of the class.*

Example:

The color and gender of a cat depends on the *instance* of the cat.

The number of legs is a property of *all* cats.

```
class Cat {
  static int NUMBER_OF_LEGS = 4;
  String gender;
  String breed;
  String color;
  boolean neutered;


}
```

Cat
NUMBER_OF_LEGS

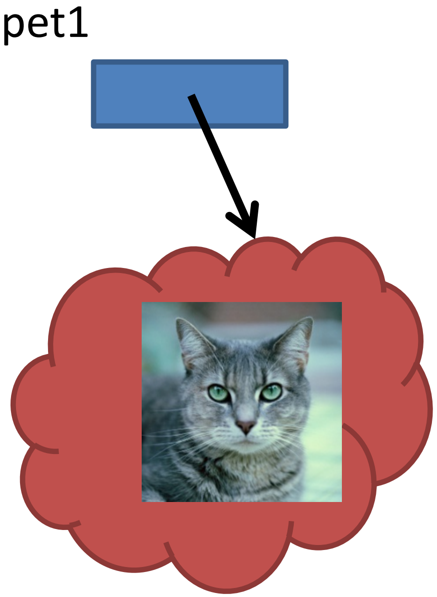
```
int leg = Cat.NUMBER_OF_LEGS;
```

NUMBER_OF_LEGS is *static* so its value comes from the class Cat

Cat
NUMBER_OF_LEGS

```
String myCatBreed = pet1.breed;
```

breed is an instance variable so its value comes from a particular instance of a Cat



Can *static* variables be changed?

Yes.

```
class Cat {
   static int numberOfCats = 0;
   static int NUMBER_OF_LEGS = 4;
   String gender;
   String breed;
   String color;
   boolean neutered;
   public Cat (…) {
        numberOfCats++;
   }//constructor
}
pet1 = new Cat(…);    //numberOfCats is now 1
pet2 = new Cat(…);    //numberOfCats is now 2
```

Use the *final* modifer to make constants.

```
class Cat {
  static int numberOfCats = 0;
  static final int NUMBER_OF_LEGS = 4;
  String gender;
  String breed;
  String color;
  boolean neutered;
public Cat (…) {
  numberOfCats++;
}
}
```

methods

methods (*functions in C++)* define the behavior of an object.

A Cat object could tell you its breed:

```
String b = pet1.getBreed();
```

```
public String getBreed() {
  return breed;
}
```

Or, if you have your cat neutered:

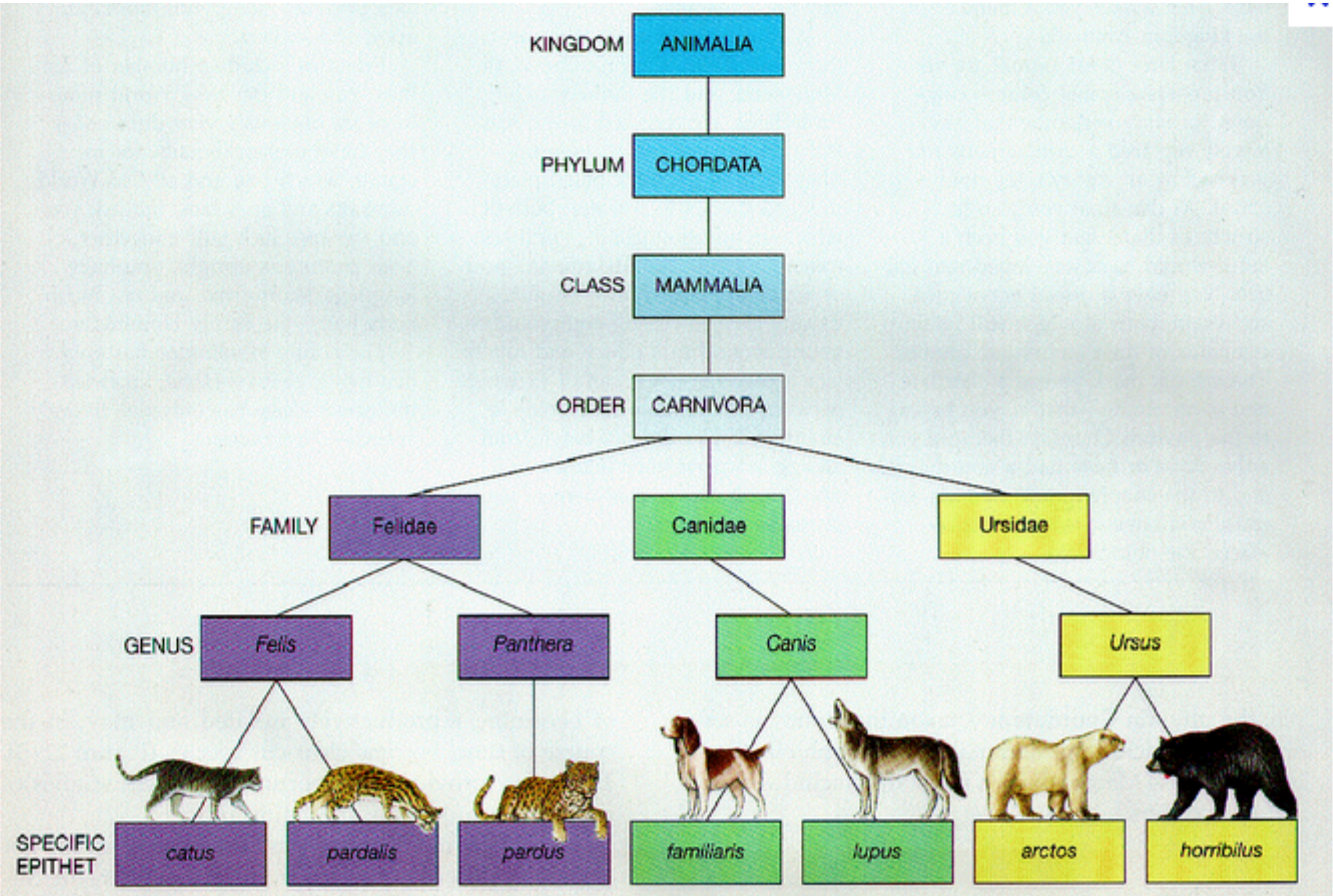
```
pet1.setNeutered();
```

```
public void setNeutered() {
  neutered = true;
}
```

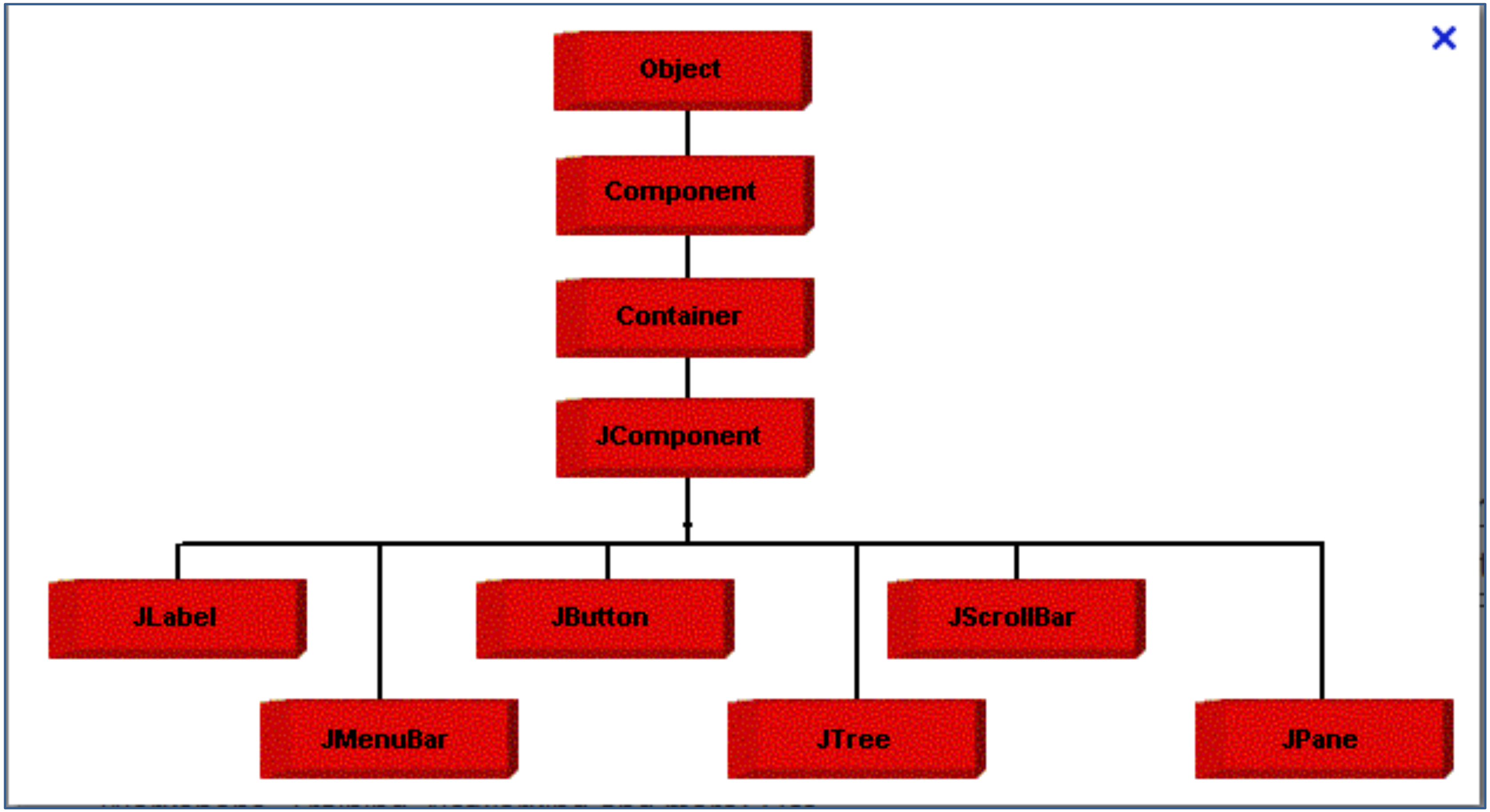
Methods can also be *static*

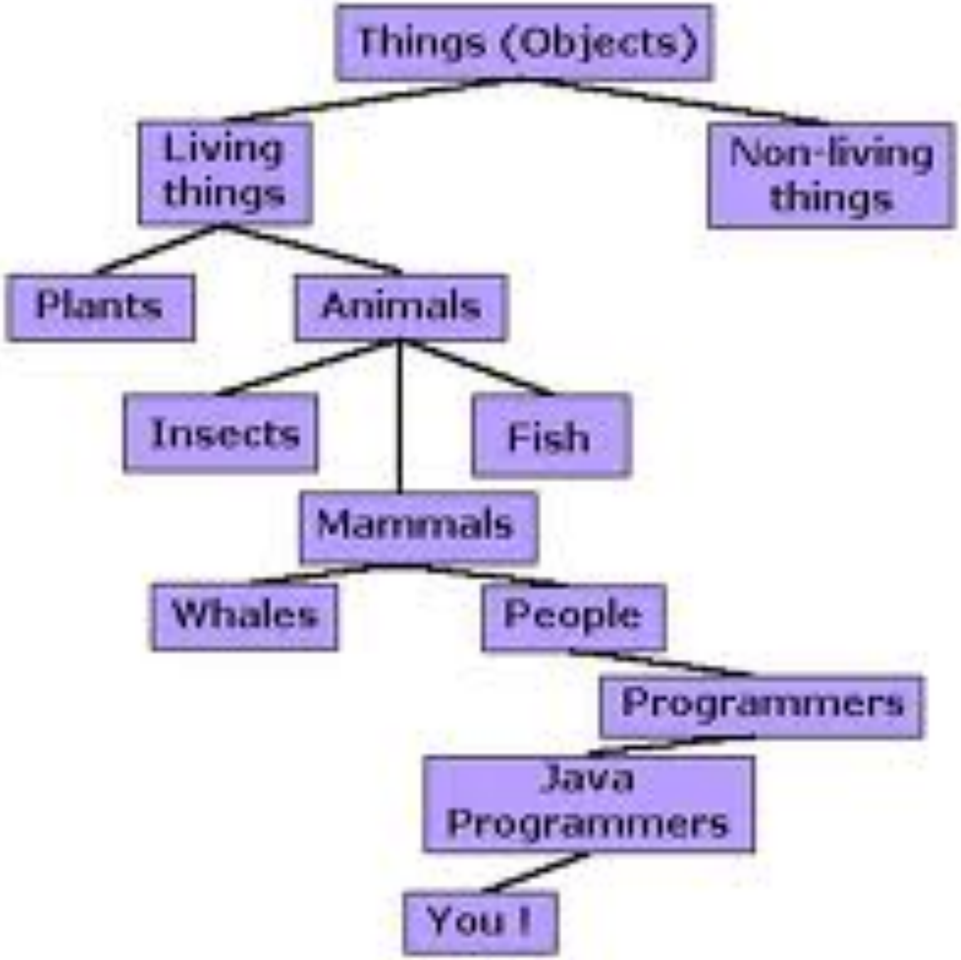
```
Cat.getNumLegs();
```

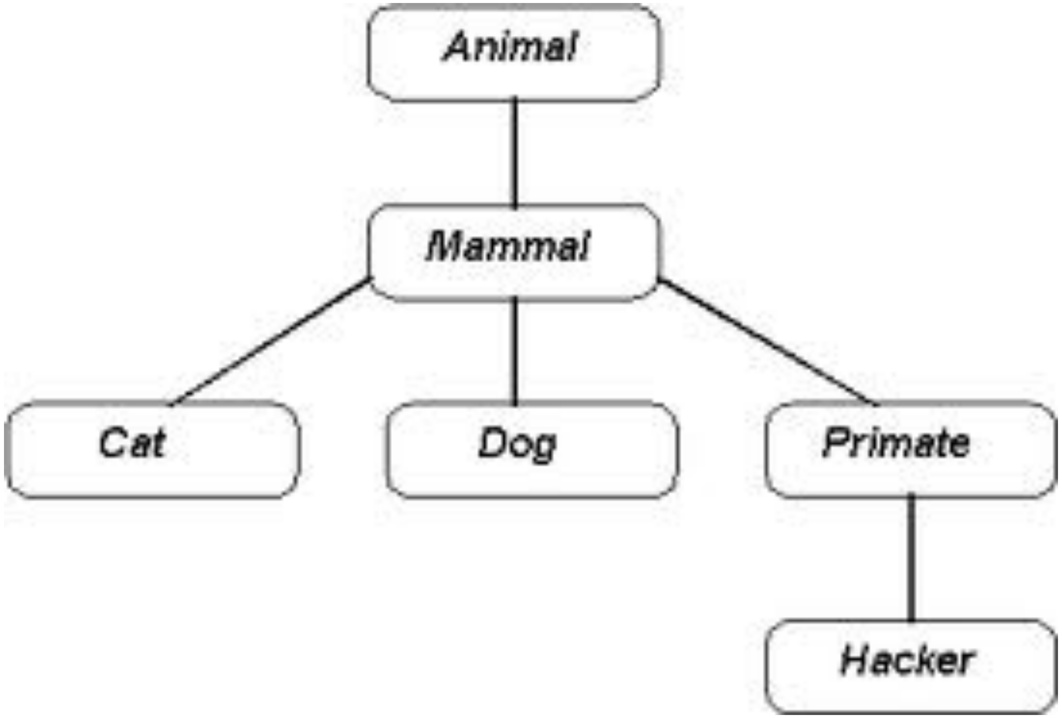
Inheritance



Example of Inheritance in the Java Class Hierarchy




Things (Objects)
Living things
Non-living things
Plants
Animals
Insects
Fish
Mammals
Whales
People
Programmers
Java Programmers
You !


Animal
Mammal
Cat
Dog
Primate
Hacker