Java Tutorials

`http://download.oracle.com/javase/tutorial/java/index.html`

Java Basics

Java Class Definition

```
public class Dog {

   Fields (attributes)

   Constructor

   Methods (operations or behaviors)

}
```


Accessibility

Classes, fields and methods must be declared with *access modifiers*.

- Classes and methods are usually declared as `public`—accessible by anyone from anywhere.
- Fields are usually declared as `private`—accessible only within the object.

Defining fields

```
public class Dog {

  private String breed;
  private char gender;
  private float height;
  private float weight;
  private int age;

  Constructor

  Methods (operations or behaviors)

}
```


Primitive data types

Fields are often defined using variables. In Java, variables must be declared with types. There are four primitive data types:

- `int`—integers: `..., -1, 0, 1, 2, ...`
- `float`, `double`—reals: `..., 0.0, ...`
- `char`—letters: `'a', 'b', 'c', ...`
- `boolean`—boolean values: `true, false`

Parameters

- Many (but not all) constructors and methods have parameters, just like many functions have parameters.
- In Java, parameters must be declared with types, just like variables must be declared with types.

Constructor

- A *constructor* is a special method that instantiates an object.
- Objects are created from a class by invoking a constructor.
- Constructors usually have parameters, and constructors create objects according to the values of parameters.

Methods

- *Methods* define objects' operations, similar to Python methods.
- Many (but not all) methods return values, just like many functions return values.
- All methods must be declared with return types (which specify the types of values being returned).

Defining a constructor

```
public class Dog {

  private String breed;
  private char gender;
  private float height;
  private float weight;
  private int age;

  public Dog(String b, char g, float h, float w, int a) {
    breed = b;
    gender = g;
    height = h;
    weight = w;
    age = a;
  }

  Methods (operations or behaviors)

}
```


Defining methods

```
public class Dog {

  private String breed;
  private char gender;
  private float height;
  private float weight;
  private int age;

  Constructor

  public void getOlder() {
    age = age + 1;
  }

}
```


Defining methods

```
public class Dog {

  private String breed;
  private char gender;
  private float height;
  private float weight;
  private int age;

  Constructor

  public int getAge() {
    return age;
  }

}
```

Accessors

- A method that simply returns the value of a field is called an *accessor* (or *"get" method*).
- An accessor for a field `f` is usually named `getF`.
- An accessor has no parameter.
- The return type of an accessor for a field `f` is the type of `f`.

Accessors

```
public class Dog {

  private String breed;

  ...

  public String getBreed() {
    return breed;
  }

}
```


Mutators

- A method that simply changes the value of a field by a parameter value is called a *mutator* (or *modifier* or *"set" method*).
- A mutator for a field `f` is usually named `setF`.
- A mutator returns nothing.
- A mutator for a field `f` has a parameter whose type is the same as that of `f`.

Mutators

```
public class Dog {

  private String breed;

  ...

  public String getBreed() {
    return breed;
  }

  public void setBreed(String breed) {
    this.breed = breed;
  }

}
```

Example: The Dog Class

http://turing.cs.trincoll.edu/cpsc115/f10/morelli/Dog.java