# Recap

**Objects** → Fields: Describe the properties of objects
→ Methods: Define what actions object can perform
→ Constructor: Defines how the object is created

Fields, methods, constructors are defined in **classes**

Classes —Define→ Object Types

Classes : Objects :: Blueprints : Houses

Public vs Private

↓ Can be accessed outside of the class

↓ Can only be accessed within a class

- Classes → public
- Fields → private
- Methods → public* OR private
- Constructors → public

## Writing Classes

Proper cased ↓

```
public class ClassName {

    // fields, constructor, methods

}
```

## Writing Fields

private DataType name; ← Leave this initialized because your Constructor will initialize it for you.

## Writing Methods

RECAP → Static functions - Methods that can run directly from a class! ↳ Do not run from an object.

Non Static

```
Car a = new Car();
a.openDoor();
```

⇒

Static

```
Car.openDoor();
```

→ Returning methods → Provides output for a method.

→ Void methods → DO NOT provide output for a method.

↳ can be private in certain cases

OPTIONAL

Return Type

Inputs

```
public static void methodName(arg1, arg2...) {
    // actions to perform
    /* return value; */ ← final return line if necessary
}
```

# Writing a Constructor

HAS to be EXACT name of class

Parameters

```
public Name of Class (param) {
    /* this.field = param; */
```

this. — Dot Operator; field — Field; = — Assign Op.

this keyword allows us to access the newly created object that is created when running the constructor.