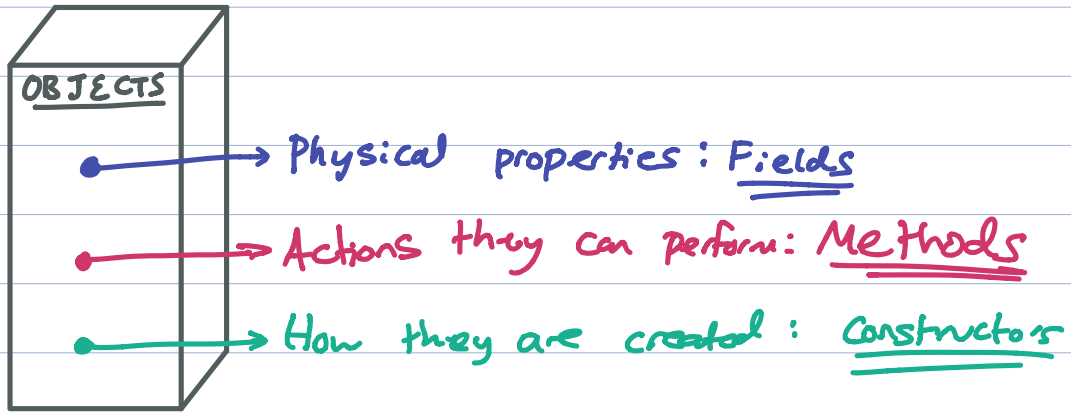
UNIT 2: Using Objects pt. I

Java is an OBJECT-oriented programing language.

Objects - Singular entity that do the actual work.



Example)

FIELD

myTaxi.maxNumOfPassengers; >>> value

METHOD

myTaxi.driveForward(inputs); >>> no value (void)

OR

>>> output (return)

CONSTRUCTOR

```
Taxi newTaxi = new Taxi(inputs);
```

- `Taxi` — reference data type
- `newTaxi` — name
- `=` — assignment operator
- `new` — new keyword
- `Taxi(inputs)` — same name as name of type!
- `;` >>> returns newly created object w/ initialized fields

Example)

```
String myString = new String("Hello!");
```

- `String` — reference type

Strings are objects, like all reference types.

Static

↳ Call it from the class names.

Ex) `Type.method()`

Math

Non-Static

↳ Called from Objects

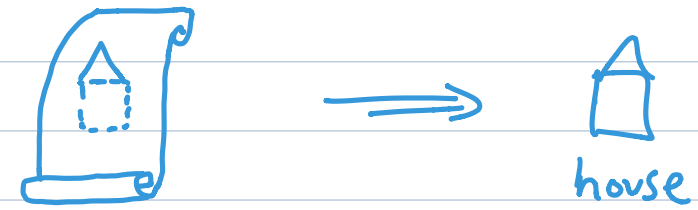
Ex)
```
Type name = new Type();
name.method()
```

String

Class - Blueprint used to define a type (for objects)

```
class TypeName {
    //code
}
```

Capital



Class ⟹ Object

available outside of file

```
public class Taxi {

    // creating variables/fields
    private int numPassengers;          // leaves variable initialized → No initial value
    private boolean isConvertible;
    private String make;   private String model;

    // methods
    public void addPassengers(int num) {        // keyword for no return value; specify inputs
        this.numPassengers += num;              // refers to this object
    }
    public int currentNumOfPassengers() {       // if does return, type of return
        return this.numPassengers;              // keyword to return a value
    }
    public void makeAndModel() {
        System.out.println(this.make + " " + this.model);
    }

    // Constructor
    public Taxi(int num, boolean isConvt, String mk, String mdl) {   // always have the same name as class
        this.numPassengers = num;
        this.isConvertible = isConvt;
        this.make = mk;  this.model = mdl;
    }
}
// end of class
```

NOT available for use outside object

available for use

UNIT 2: Using Objects pt. II

java.lang

String class

methods

`.length();`
↳ Returns number of characters in main String.

`.indexOf(string);`
↳ Returns location of a string in main string.

`.substring(start, end);` ← optional
↳ Returns string in original string based on range

`.compareTo(string);`
↳ Returns value based on string comparison.

String Indexes

Always start at 0

H e l l o !
0 1 2 3 4 5

Math class

Static methods

`.abs(num);`
↳ Returns absolute value of input number.

`.pow(num1 (base), num2 (exponent));`
↳ Returns num1 raised to the power of num2

`.sqrt(num);`
↳ Returns square root of input number.

`.random();`
↳ Returns random # between 0 and 1 (as a double)

0! 1 2 3 4 5

Ex)

String myString = "Hello";
0 1 2 3 4

myString.length ( ); >>> 5

myString.indexOf("e"); >>> 1

myString.indexOf("a"); >>> -1

myString.substring(2, 3); >>> l

myString.compareTo("Bye"); >>> 6

Ex)

Math.abs (-8.2);
>>> |-8.2| = 8.2

Math.pwr (2, 3);
>>> $2^3 \rightarrow 8$

Math.sqrt (9);
>>> $\sqrt{9} \rightarrow 3$

Math.random ( );
>>> 0.9052567...

"My String"

"My String is "Ajay""

Escape Characters

\"	\n	\t	\\
quote	line	tab	backslash

"My String is "Ajay""