CSCI 136: Data Structures and Advanced Programming

Lecture 3

Java Crash Course

Instructor: Dan Barowy

Williams

Quiz

Outline

Study tip

Coinstrip

Personal computers

Java crash course

Program design

Life skill #3: growth mindset

Have you ever thought:
"I'm not good at [x]"

Life skill #3: growth mindset

If you are motivated and study effectively, there is nothing you cannot learn.

In fact, you learn whether you want to or not.

Proof (demo).

Every brain is an amazing learning machine. Learn how to use it!

Coinstrip demo in lab today

Personal computer use

Java crash course

Input

1. Static input (constants)
2. Dynamic input
   1. **`args`**
   2. **`Scanner`**

differences?

`Scanner` automatically converts type

`Scanner` is interactive

`Scanner` example (code)

Static types

Static types

A **type system** is a **set of rules** that assigns a property called type to values used in a computer program. Types enforce the otherwise implicit categories the programmer uses for data (e.g., "number", "word", "picture"). The main purpose of a type system is **to reduce the number and kind of bugs** in computer programs by ensuring that values are used consistently.

```
int x = 34;          // OK
int x = 3.4;         // Not OK
int x = "Dan";       // Not OK
```

Q: Why do we call them “static” types?

```
class Foo {
  int x = “Dan”;
}
```

```
Foo.java:2: error: incompatible types: String cannot be
converted to int
    int x = "Dan";
            ^
1 error
```

The compiler is your new best friend.

It tells you what’s wrong and where!

Always be compiling (ABC)!

Q: Where else have we seen “static”?

```
public static void main(String[] args) {
    System.out.println(“Pay attention!”);
}
```

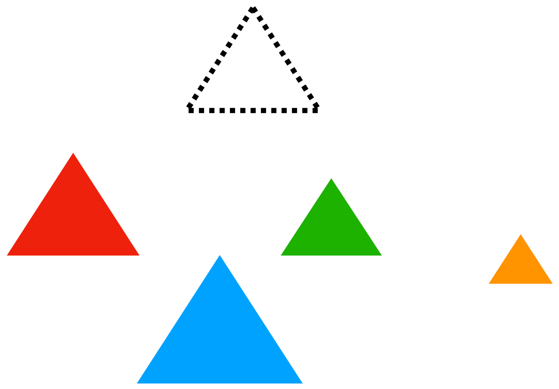
Classes and objects

Classes

A **class** is a form of **data abstraction**. The purpose of a class is to separate the details that are important to the programmer (the **interface**) from the details that are important to the computer (the **implementation**). Classes are a key building block in designing data structures.

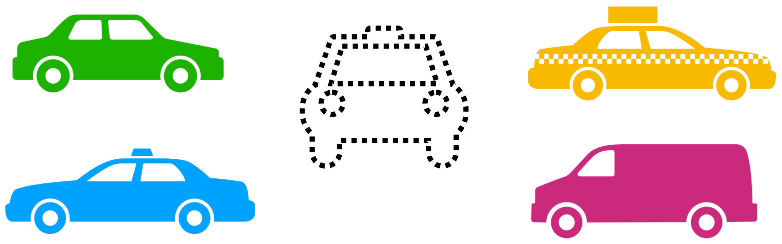
Classes are prototypes.

Objects are copies ("instances").



"Car" is a prototype.

There are many instances of cars.



All cars have the same interface.

(wheels, doors, steering wheel, etc.)

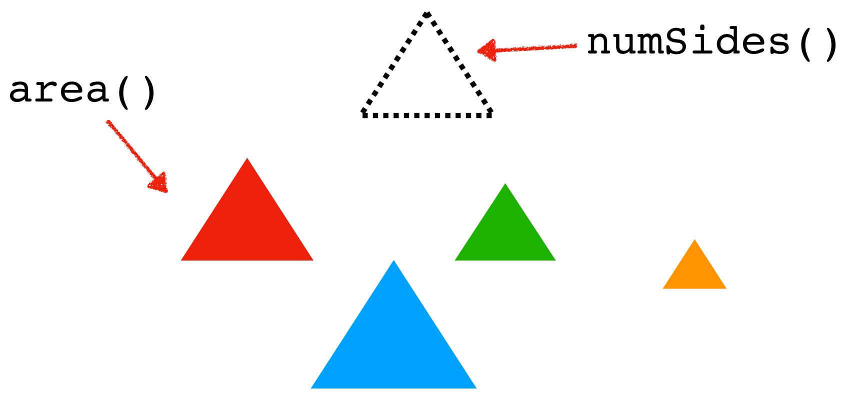
```
public static void main(String[] args) {
    System.out.println("Pay attention!");
}
```

Methods are functions that are tied to either:
1. a class, or ← static method
2. an instance of a class (an object).

instance method → object

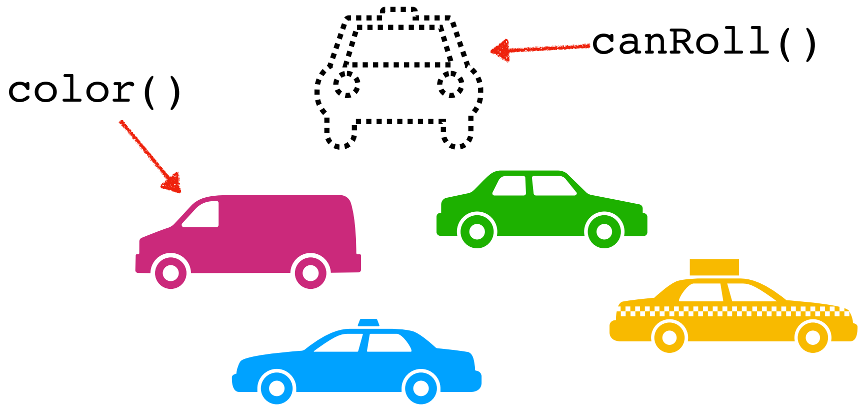
static methods are "attached" to class.

instance methods are "attached" to object.



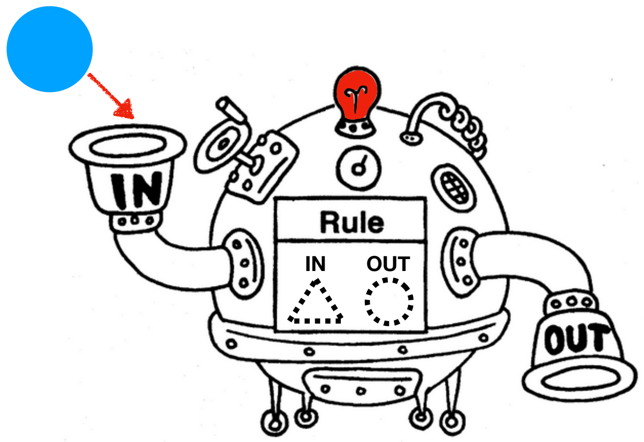
static methods are "attached" to class.

instance methods are "attached" to object.



A class also defines a static type.

Using object incorrectly yields a type error.



Q: How might we represent a sequence of words using a class?

Let's sketch out a design.

Recap & Next Class

Today we learned:

- More Java
  - More I/O
  - Types
  - Classes
- Design documents

Next class:

- Implementation of design
- Version control
- Generics