CSCI 334:
Principles of Programming Languages

Lecture 20: Object-oriented programming II

Instructor: Dan Barowy

Williams

Topics

Resume project activity

Dynamic dispatch

C++ (time permitting)

Activity

Read your partners' project proposal (10 minutes).

Take turns coming up with a program in their language *on paper*.

Partner: does this program make sense? If not, what about your documentation tripped them up?

OO vs Functional Tradeoff

- OO offers a different kind of extensibility than functional (or function-oriented) languages.
- Suppose you're modeling a hospital.

Operation	Doctor	Nurse	Orderly
`Print`	`Print Doctor`	`Print Nurse`	`Print Orderly`
`Pay`	`Pay Doctor`	`Pay Nurse`	`Pay Orderly`

- FP makes it easy to add operations (rows).
- OOP makes it easy to add data (columns).

Dynamic Dispatch

(the secret to understanding how
Java, Python, Ruby, etc. work)

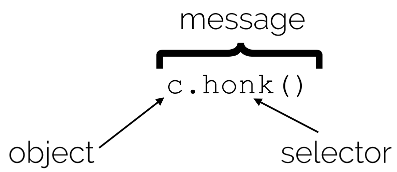
Ingalls Test for Extensibility

- The test is about the ability to extend software *after* it has already been designed and written.
- E.g., suppose you have a class for a `ColoredRectangle`.
- Can you define a new kind of number (e.g., fractions), use your new numbers to redefine (subtype) rectangle, and then ask the system to color the rectangle?
- If so, you have an OO system.

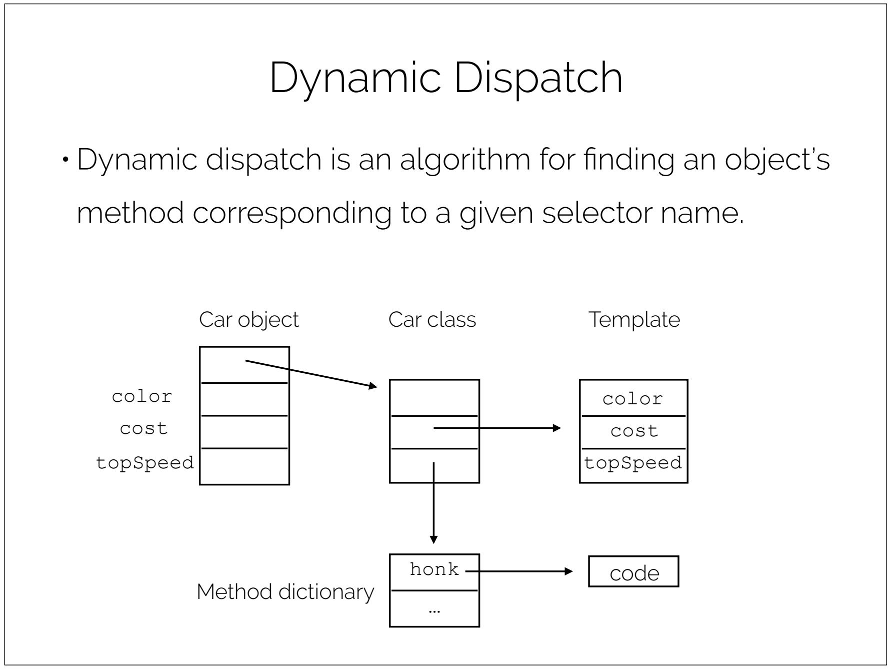
Dynamic Dispatch Example

Dynamic Dispatch

- Dynamic dispatch is the OO mechanism for polymorphism.
- Functions ("methods") are always bound to an object (or class).
- A method is called ("dispatched") by sending a "message" to the "selector" of an object.

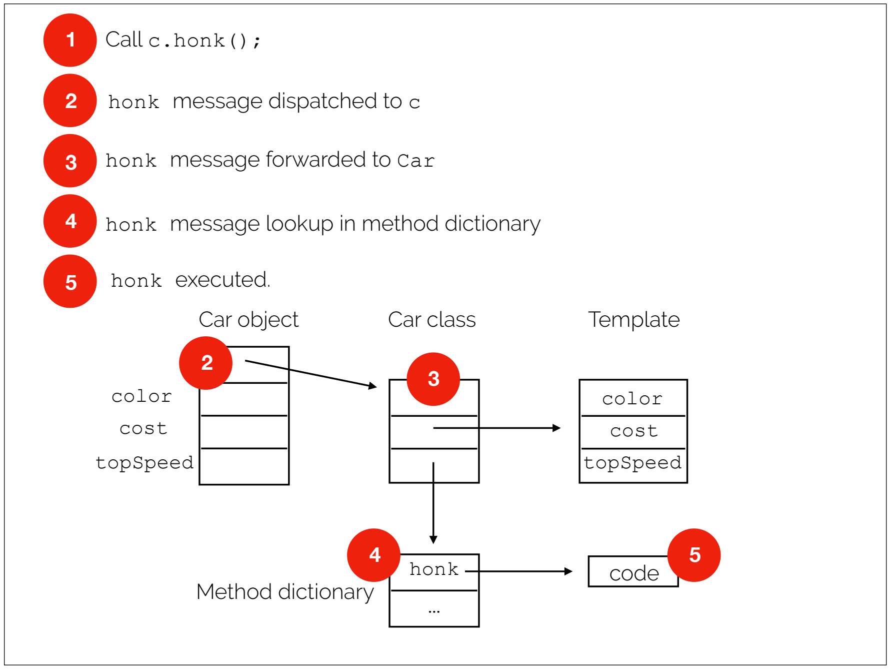


```
c.honk()
```



Dynamic Dispatch

- Dynamic dispatch is an algorithm for finding an object's method corresponding to a given selector name.



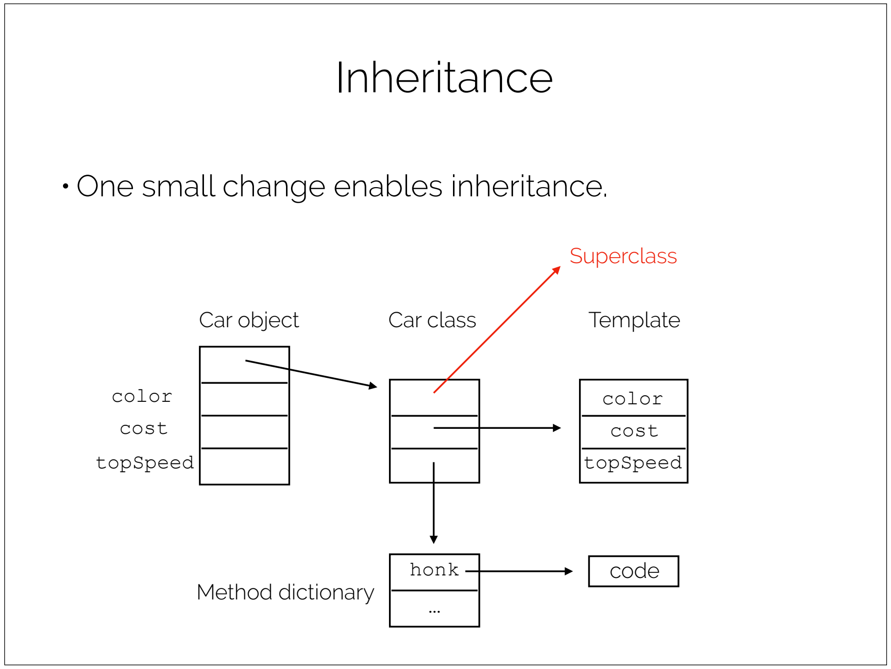
1 Call `c.honk();`

2 `honk` message dispatched to `c`

3 `honk` message forwarded to `Car`

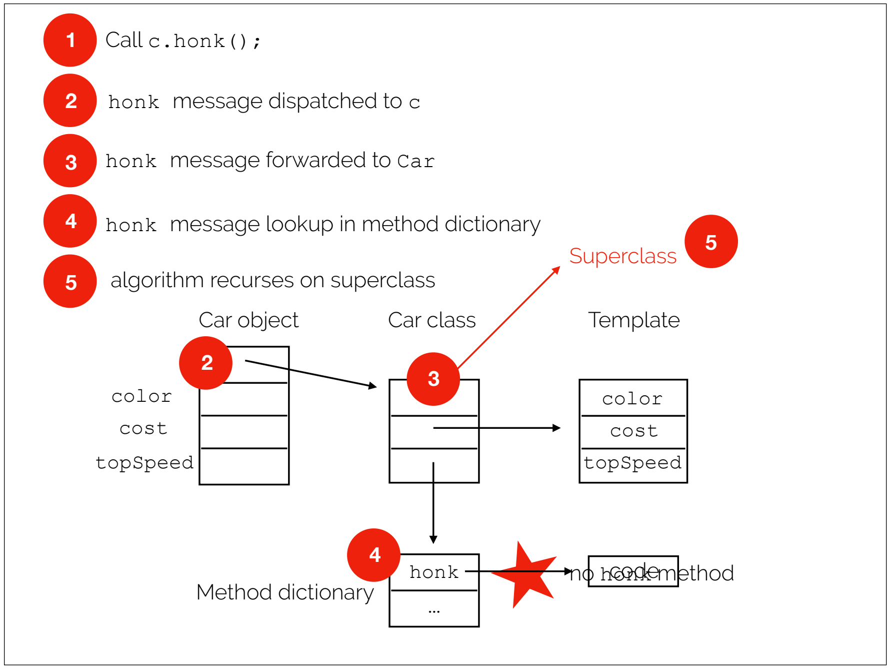
4 `honk` message lookup in method dictionary

5 `honk` executed.



Inheritance

- One small change enables inheritance.



1 Call `c.honk();`

2 `honk` message dispatched to `c`

3 `honk` message forwarded to `Car`

4 `honk` message lookup in method dictionary

5 algorithm recurses on superclass

Activity

Draw the dynamic dispatch data structures for `Honks`, `Car`, and `Goose`.

C++

Efficient object oriented programming.

History of C++

- Began originally in 1979 with Bjarne Stroustrup's "C with Classes"
- C++ released in 1983 with most of the major features we know today.
- Design was strongly influenced by Simula, but Simula was too slow. Stroustrup wanted a fast, portable, language with object-oriented features. C had everything but OO.
- C++ is largely a superset of C. Until C++98, every C program was a valid C++ program. Still relatively easy to convert C to C++.
- Major driving philosophy: "only pay for what you use."
- C++ has many features. We will cover only the essential ones here.

C++ example