Northeastern University

Lecture 5: Java Review - 5 (Self-Study)

Prof. Chen-Hsiang (Jones) Yu, Ph.D.

College of Engineering

Materials are edited by Prof. Jones Yu from

Data Structures and Abstractions with Java, 5th edition. By Frank M. Carrano and Timothy M. Henry.
ISBN-13 978-0-13-483169-5 © 2019 Pearson Education, Inc.

JAVA: An Introduction to Problem Solving & Programming, 7th Ed. By Walter Savitch
ISBN 0133766268 © 2014 Pearson Education, Inc.

What have we reviewed?

- Encapsulation & Abstraction
- Methods
- Interfaces
- System & Classes Design (UML)
- Classes, Methods, Objects, Inheritance (Labs)

Outline

- Classes and Methods (Lecture 5)
- Objects and Methods (Lecture 6)
  - » Constructors
  - » Overloading

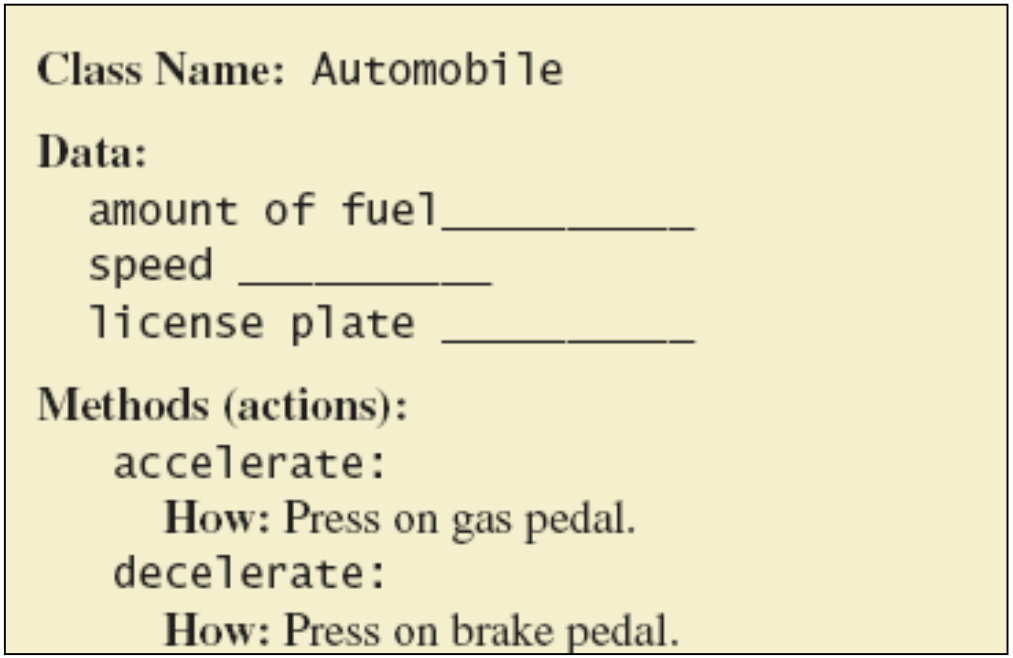
Classes and Methods

Class and Method Definitions

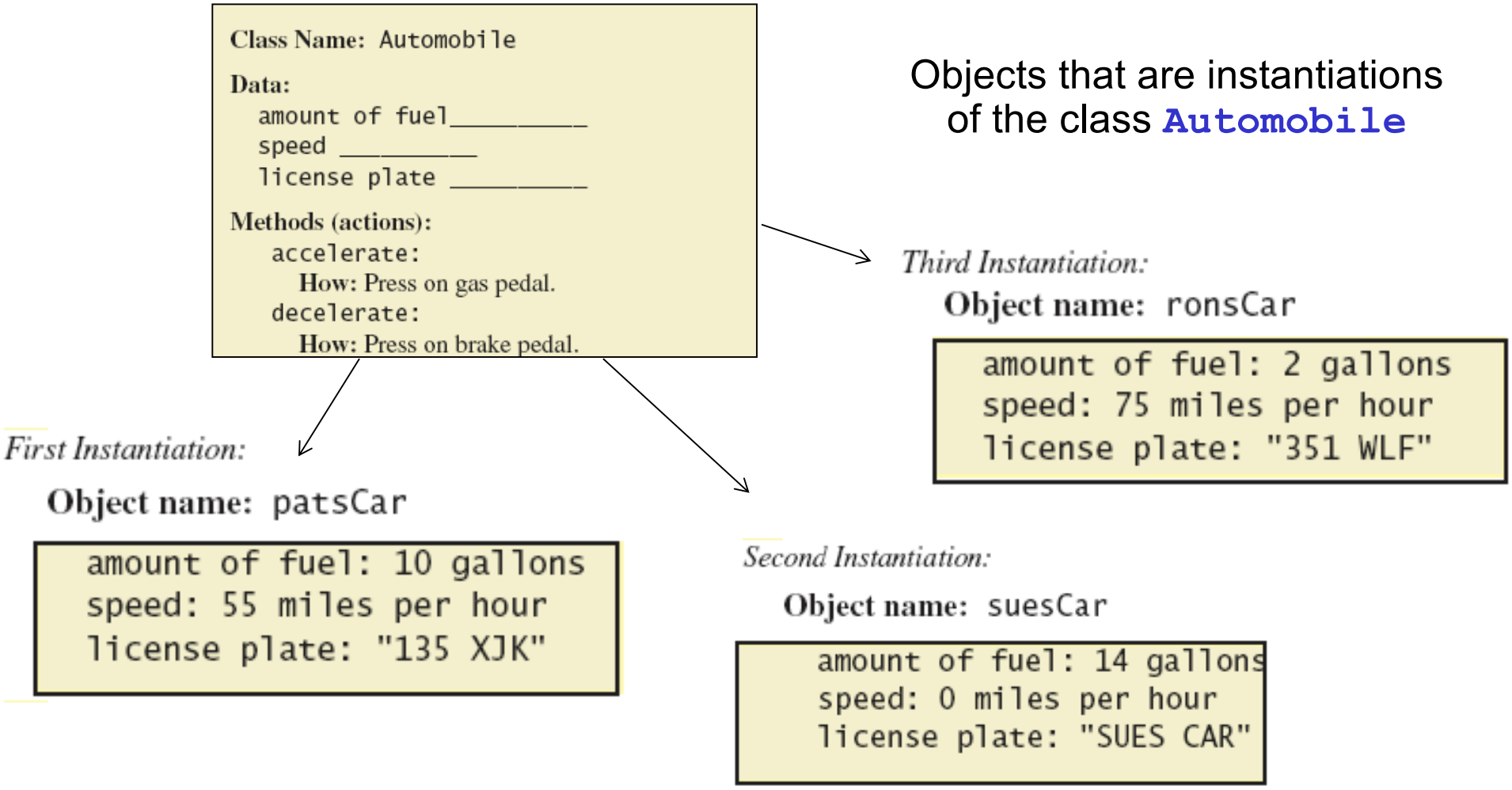
- Java program consists of objects
  - » Objects of Class types
  - » Objects that interact with one another
- Program objects can represent
  - » Objects in real world
  - » Abstractions

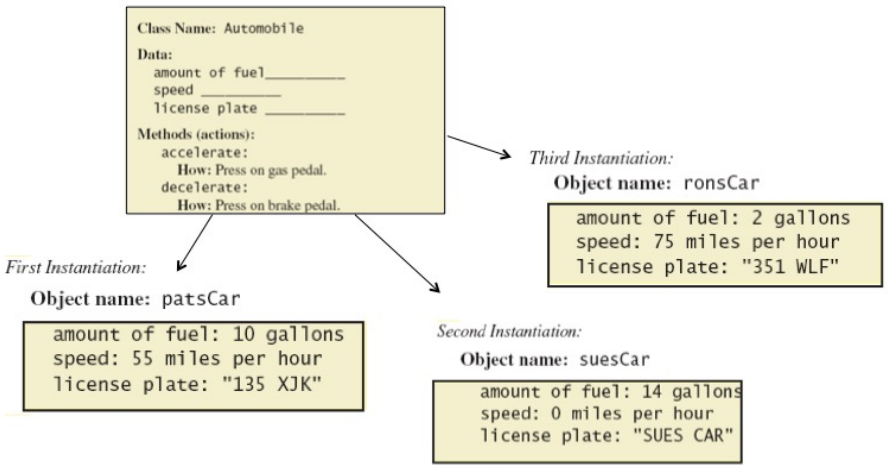
Class and Method Definitions

- A class as a blueprint



Class and Method Definitions





Java Project

Automobile.java

CarSystem.java

Objects

- patsCar
- suesCar
- ronsCar

```
Automobile patsCar = new Automobile();
```

or

```
Automobile patsCar =
     new Automobile(10, 55, "135 XJK");
```

Class Files and Separate Compilation

- Each `Java` class definition usually in a file by itself
  - » File begins with name of the class
  - » Ends with `.java`
- Class can be compiled separately
- Helpful to keep all class files used by a program in the same directory

`Dog` class and Instance Variables

- View sample program - `class Dog`
- `Dog` class has
  - » Three pieces of data (instance variables)
  - » Two behaviors
- Each instance of this type has its own copies of the data items
- Use of `public`
  - » No restrictions on how variables used

```
public class Dog
{
    public String name;
    public String breed;
    public int age;
    public void writeOutput()
    {
        System.out.println("Name: " + name);
        System.out.println("Breed: " + breed);
        System.out.println("Age in calendar years: " +
                           age);
        System.out.println("Age in human years: " +
                           getAgeInHumanYears());
        System.out.println();
    }
    public int getAgeInHumanYears()
    {
        int humanAge = 0;
        if (age <= 2)
        {
            humanAge = age * 11;
        }
        else
        {
            humanAge = 22 + ((age-2) * 5);
        }
            return humanAge;
    }
}
```

Using a Class and Its Methods

- View sample program

  `class DogDemo`

Sample screen output

```
Name: Balto
Breed: Siberian Husky
Age in calendar years: 8
Age in human years: 52

Scooby is a Great Dane.
He is 42 years old, or 222 in human years.
```

```
public class DogDemo
{
    public static void main(String[] args)
    {
        Dog balto = new Dog();
        balto.name = "Balto";
        balto.age = 8;
        balto.breed = "Siberian Husky";
        balto.writeOutput();

        Dog scooby = new Dog();
        scooby.name = "Scooby";
        scooby.age = 42;
        scooby.breed = "Great Dane";
        System.out.println(scooby.name + " is a " +
                           scooby.breed + ".");
        System.out.print("He is " + scooby.age +
                         " years old, or ");
        int humanYears = scooby.getAgeInHumanYears();
        System.out.println(humanYears + " in human years.");
    }
}
```

Sample Screen Output

```
Name: Balto
Breed: Siberian Husky
Age in calendar years: 8
Age in human years: 52

Scooby is a Great Dane.
He is 42 years old, or 222 in human years.
```

Exercise

- Give the complete definition of a class called Person that has two instance variables, one for the person's name and the other for the person's age.
- Include accessor methods (getters) and mutator methods (setters) for each instance variable.
- Also, include a method that sets both the name and age of a person.

Answer

```
public class Person {
     private String name;
     private int age;

     public void setName(String newName){
          name = newName;
     }

     public void setAge(int newAge){
          if(newAge > 0){
               age = newAge;
          }else{
               System.out.println("Error: Age is negative");
               System.exit(0);
          }
     }

     public void setPerson(String newName, int newAge){
          setName(newName);
          setAge(newAge);
     }

     public String getName(){
          return name;
     }

     public int getAge(){
          return age;
     }
}
```

The Keyword `this`

- Referring to instance variables outside the class – must use
  - » Name of an object of the class
  - » Followed by a dot
  - » Name of instance variable
- Inside the class,
  - » Use name of variable alone
  - » The object (unnamed) is understood to be there

The Keyword `this`

- Inside the class the unnamed object can be referred to with the name `this`
- Example

```
this.name = keyboard.nextLine();
```

- The keyword `this` stands for the receiving object

Exercise

- Create an `Automobile` Class to represent cars. It contains attributes: amount of fuels, speed and license plate.
- Write a test program to verify the implementation of the `Automobile` Class.
  - » Specifically, create 3 objects:
    - patsCar: 10 gallons, 55 miles per hour, “135 XJK”
    - sueCar: 14 gallons, 30 miles per hour, “SUES CAR”
    - ronsCar: 2 gallons, 35 miles per hour, “351 WLF”
  - » Print out the information of all cars (using toString())

Answer

```
public class Automobile {
    private int amountFuel;
    private int speed;
    private String licensePlate;

    public Automobile(int a, int s, String l){
        this.amountFuel = a;
        this.speed = s;
        this.licensePlate = l;
    }

    public String toString(){
        return "Amount of fuel = " + amountFuel + "\nSpeed = "
                + speed + "\nlicensePlate = " + licensePlate;
    }
}
```

Automobile.java

Answer

```
public class AutomobileDemo {

  public static void main(String[] args){
    Automobile patsCar = new Automobile(10, 55, "135 XJK");
    Automobile suesCar = new Automobile(14, 0, "SUES CAR");
    Automobile ronsCar = new Automobile(2, 75, "351 WLF");

    System.out.println(patsCar);
    System.out.println("");
    System.out.println(suesCar);
    System.out.println("");
    System.out.println(ronsCar);
  }

}
```

AutomobileDemo.java

Answer

```
Amount of fuel = 10
Speed = 55
licensePlate = 135 XJK

Amount of fuel = 14
Speed = 0
licensePlate = SUES CAR

Amount of fuel = 2
Speed = 75
licensePlate = 351 WLF
```

Output Results