COMP90041

Programming and Software Development

Semester 2, 2021

Lab 4

Andrew Naughton

andrew.naughton@unimelb.edu.au

Dog class

Classes and objects

- Each object is an instance of some class

Template / blueprint !

A class holds operations and data related to one concept.

- When creating an object, its instance variables need to be initialised to appropriate values
- Constructors are special methods responsible for this

```
public ClassName(type1 var1,...) {
    ⋮
}
```

```
Classname myObject = new Classname(…);
```

Calls / invokes

Public and private modifiers

public: no restrictions on access; vulnerable to outside interference/use

private: cannot be accessed by name outside the class

Always make variables and methods private unless there is a need or good design reason not to (rare)

Variables

- Instance variables, which hold the data of an object

Form: `private` *`type name`* `;`

```
public class Person {
    private String familyName;
    private String givenName;
    ⋮
}
```

static

- Local variables live in a method; class variables live in a class; instance variables live in an object

Methods

- (Instance) methods, which define the operations (code) of an object

```
public class SampleClass {

    public static void method1(){
        method2();
    }

    public void method2(){
        method1();
    }
}
```

Call a static method

```
SampleClass.method1();
```

Call a non-static method

```
SampleClass myObject = new SampleClass();

myObject.method2;
```

Headers and signatures

- First part of method definition (up to {) is called the method header
- Method name plus number and types of arguments together are called the method signature

Header

```
public static int calInt(int num1, int num2){
    return num1 + num2;
}
```

Signature

Method overloading

- Overloading: when a method name has multiple definitions, each with different signature
- Java automatically selects the method whose signature matches the call
- You cannot overload based on return type, only parameter types

Wrong!

```
int    bad(int x, double y) {...}
double bad(double x, int y) {...}
```

Right!

```
public void setDate(int month, int day,int year)
public void setDate(String month, int day,int year)
public void setDate(int year)
```

toString() method

If p is a Person object

- What should `System.out.print(p)` print?
- Define a public method `String toString()`

```
public String toString() {
    return givenName + " " + familyName;
}
```

toString() in Person class

Getters and setters

```java
public class Person {
  private String name; // private = restricted access

  // Getter
  public String getName() {
    return name;
  }

  // Setter
  public void setName(String newName) {
    this.name = newName;
  }
}
```