

Object Oriented Programming

Classes and Objects

Mark Kim

San Francisco State University

3/9/2024

Classes

Classes are an **abstract** concept. They are a blueprint for something that belongs to that class.

Example

In the following class, any person who is part of the Student class has an `id` and a `name`.

```
1 class Student {  
2     String student_id;  
3     String name;  
4 }
```

Objects

Objects are the **manifestation** of a Class.

Example

`student0` is a Student. This Student therefore has a `student_id` and a name.

This object can now be **instantiated**.

Example

```
1 Student student_0 = new Student();
```

Coding Classes and Objects

Classes are created in their own .java file.
These classes can then be used within other classes.

Example (student.java)

```
1 class Student{
2     public String id;
3     public String name;
4
5     public Student() {
6         this.id = "000000000";
7         this.name = "Default Student";
8     }
9 }
```

Coding Classes and Objects, continued

The class containing the `main` method can then instantiate objects from these classes.

Example (createStudent.java)

```
1 class StudentDriver{
2     public static void main(String[] args) {
3         Student studentA = new Student();
4     }
5
6     System.out.println(studentA.id);
7     // prints "000000000" to console
8     System.out.println(studentA.name);
9     // prints "Default Student" to console
10 }
```

Class attributes and behaviors

Earlier, our student class only had attributes (and a constructor).
Classes also have behaviors.

Example (student.java)

```
1 class Student{
2     // attributes
3     public String id;
4     public String name;
5
6     // behaviors
7     public Student() { // Constructor
8         this.id = "000000000";
9         this.name = "Default Student";
10    }
11    public compareName(Student b) {
12        return this.name.compareTo(b.name);
13    }
14 }
```