Intro to Classes

Upon completion of this module, a student will be able to

- understand an explain the difference between classes and objects
- understand what class members are
- understand, define and use constructors
- understand what magic numbers are and NEVER use them
- understand and know when to make members private or public

Assignment

- Task
  - For this project you'll build an app that is a basic contacts manager app. It will allow you to save and view contact information for your friends.
- Repo
  - https://github.com/LambdaSchool/Android_Classes
- Submission
  - Compress the project directory into a zip archive and then send it to your PM in a DM
- Challenge
  - If you finish and want another challenge try adding the ability to delete contacts.

A Student Can

understand an explain the difference between classes and objects

Classes and Objects

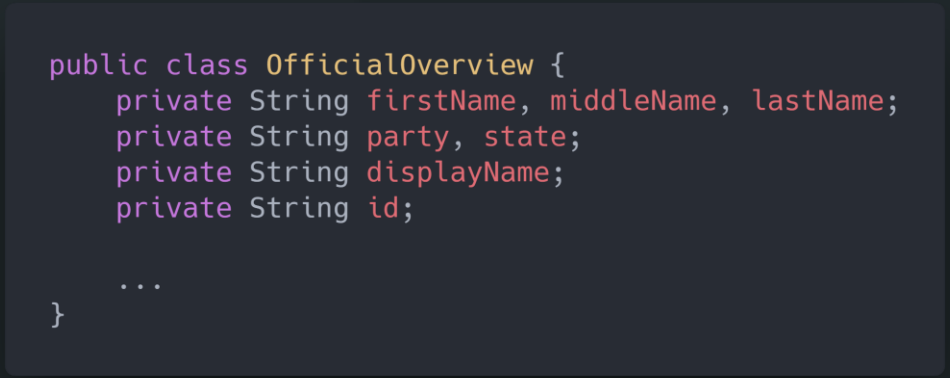
- Class
  - Blueprint for an Object
  - Can access static members

- Object
  - Instantiated class
  - Multiple instances
  - Access all public members

A Student Can

understand what class members are

Data Members

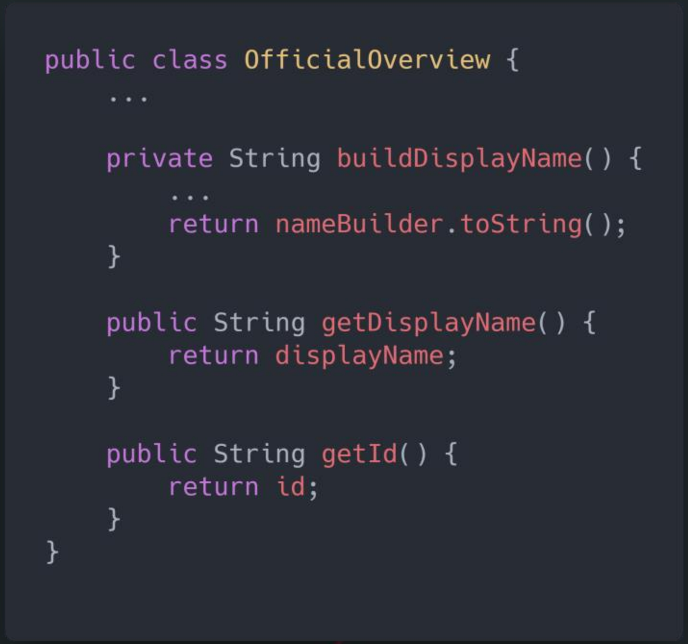


```java
public class OfficialOverview {
    private String firstName, middleName, lastName;
    private String party, state;
    private String displayName;
    private String id;

    ...
}
```

- Variables associated with a single object
    - one copy per object

Member Methods

- Specific to the class
- Access to all class members



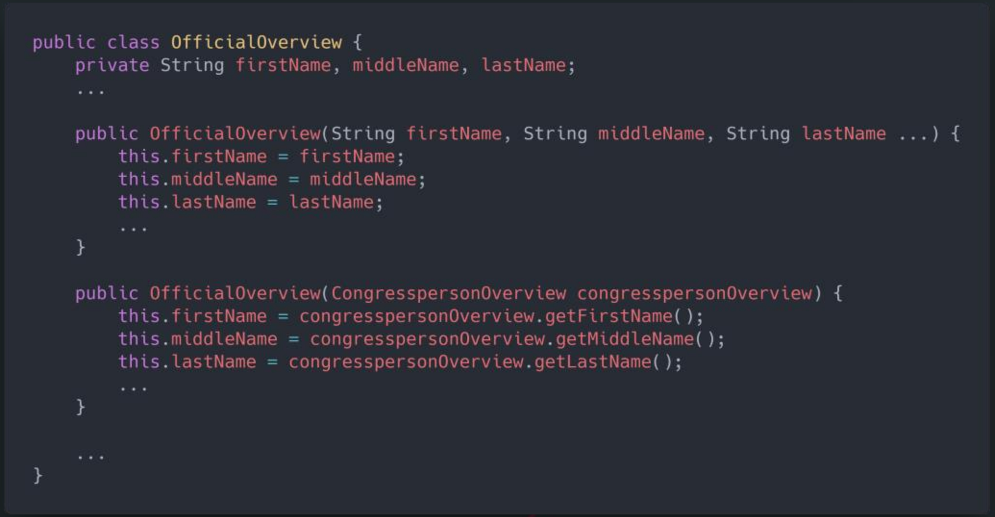
```
public class OfficialOverview {
    ...

    private String buildDisplayName() {
        ...
        return nameBuilder.toString();
    }

    public String getDisplayName() {
        return displayName;
    }

    public String getId() {
        return id;
    }
}
```

A Student Can

understand, define and use constructors

Constructors

- Called with ‘new’ keyword
- Can be overloaded
- Can call one constructor from another

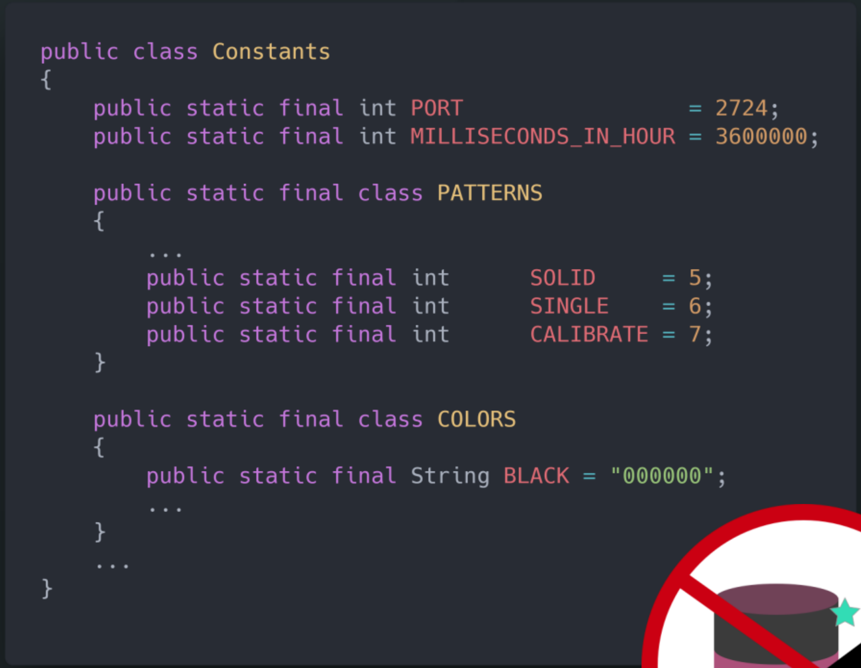


```java
public class OfficialOverview {
    private String firstName, middleName, lastName;
    ...

    public OfficialOverview(String firstName, String middleName, String lastName ...) {
        this.firstName = firstName;
        this.middleName = middleName;
        this.lastName = lastName;
        ...
    }

    public OfficialOverview(CongresspersonOverview congresspersonOverview) {
        this.firstName = congresspersonOverview.getFirstName();
        this.middleName = congresspersonOverview.getMiddleName();
        this.lastName = congresspersonOverview.getLastName();
        ...
    }

    ...
}
```

A Student Can

understand what magic numbers are and NEVER use them

Magic Numbers



```
public class Constants
{
    public static final int PORT                 = 2724;
    public static final int MILLISECONDS_IN_HOUR = 3600000;

    public static final class PATTERNS
    {
        ...
        public static final int      SOLID     = 5;
        public static final int      SINGLE    = 6;
        public static final int      CALIBRATE = 7;
    }

    public static final class COLORS
    {
        public static final String BLACK = "000000";
        ...
    }
    ...
}
```

- Numbers that don’t have name
- Define as constants
  - UPPERCASE_WITH_UNDERSCORE
  - Constants file
    - Static class
    - Only static data members
  - Class File
- “static final”

A Student Can

understand and know when to make members private or public

Public vs Private Members

- Public - visible to all
- Private - visible to current class only
- Restrict view into apps
- Private Data Members
  - Getters and Setters