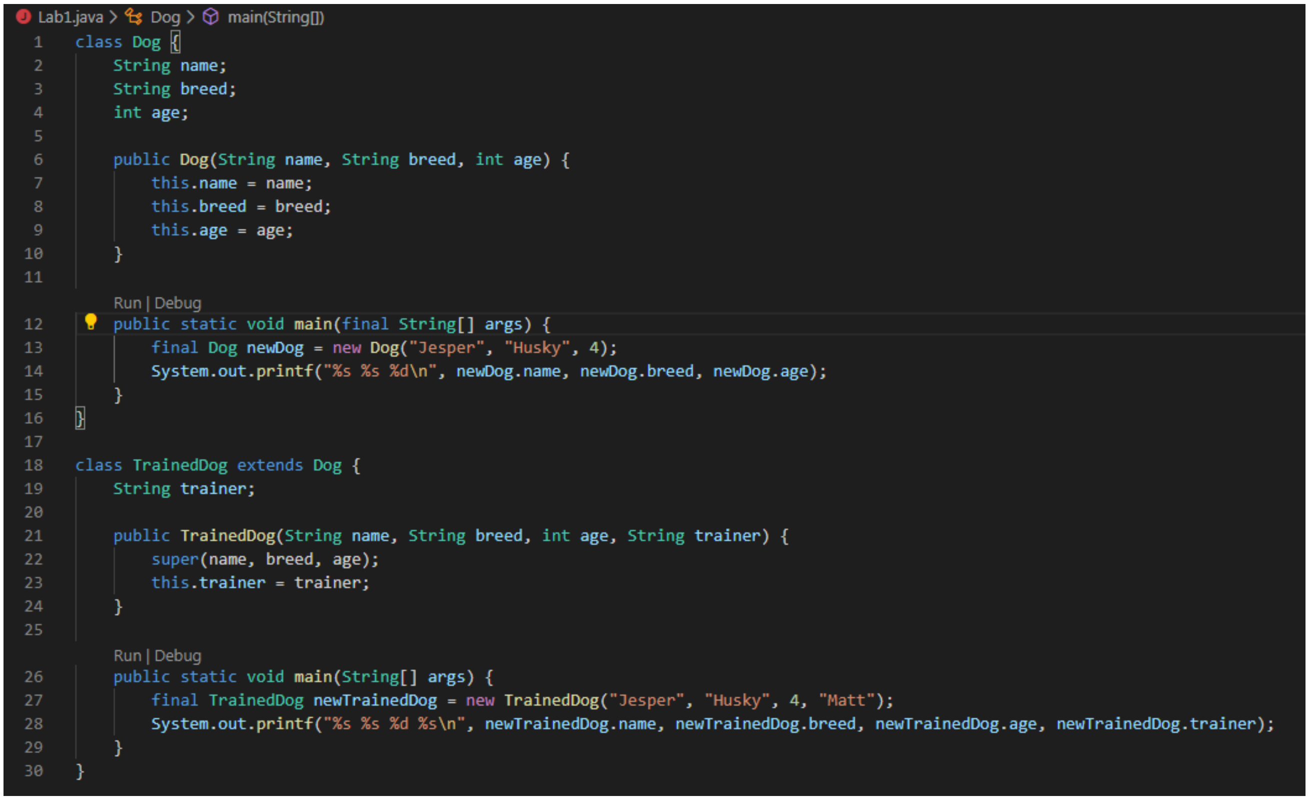
Advanced Android Programming Lab 1

1. **Object** --- Object is an instance of a class. Objects have state and behaviours. The example creates a new dog and logs it to the console.
2. **Class** --- A blueprint for an object. Describes the state and behaviour of an object. The example above works for this too. I'm just using the class Dog as a blueprint for the newDog created in the main function.
3. **Instantiation of an object** --- Creation of a new object. Usually you can use the keyword *new* to create an object. Example above works for this one too.
4. **Visibility** --- Public variables and methods are visible outside of the class they are defined in, private one's are only visible to the class they are defined in and protected one's are visible to all classes in the package and all subclasses inside or outside the package.
5. **Member datas / methods** --- There are two types of data members in Java, instance data members and static data members. Instance data members are those whose memory space is always created when a new object gets created. Static data members are those whose memory space is created only once when the class is loaded in the main memory.
6. **Inheritance** --- Means that a child class can quite literally inherit the properties of its' parent. Usually you can use the keyword *extends* to inherit the properties of a parent class.
7. **Interface** --- A collection of abstract methods. A class can implement an interface and inherit its' methods.
8. **Polymorphism** --- The ability for an object to take on various forms. Most common type of polymorphism is when a child class extends a parent class.
9. **Overriding** --- Quite literally just overriding an already existing method for example.
10. **Abstract classes** --- Classes that cannot be instantiated but can be subclassed. Abstract classes are similar to interfaces.



```java
class Dog {
    String name;
    String breed;
    int age;

    public Dog(String name, String breed, int age) {
        this.name = name;
        this.breed = breed;
        this.age = age;
    }

    public static void main(final String[] args) {
        final Dog newDog = new Dog("Jesper", "Husky", 4);
        System.out.printf("%s %s %d\n", newDog.name, newDog.breed, newDog.age);
    }
}

class TrainedDog extends Dog {
    String trainer;

    public TrainedDog(String name, String breed, int age, String trainer) {
        super(name, breed, age);
        this.trainer = trainer;
    }

    public static void main(String[] args) {
        final TrainedDog newTrainedDog = new TrainedDog("Jesper", "Husky", 4, "Matt");
        System.out.printf("%s %s %d %s\n", newTrainedDog.name, newTrainedDog.breed, newTrainedDog.age, newTrainedDog.trainer);
    }
}
```

An example for object, class, instantiation, visibility, data members, polymorphism and inheritance.

1. Android apps can be written in Java, Kotlin and C++.
2. APK is an android package file and it contains all the contents of an android application.
3. Android runs application in their own security sandbox, protected by android's security features.
4.
    - **Activities**: An activity is a component that keeps track of what the user is doing. Activities have a UI and is anything the user sees on the screen. Usually apps have at least one activity but more often than not they have more than one.
    - **Services**: Services run in the background doing performing long-running operations. Services do not have UIs as they are only meant to run in the background.
    - **Broadcast receivers**: A broadcast receiver allows the app to respond to system-wide broadcast announcements. The system can even deliver broadcasts to apps that aren't currently running. Broadcast receivers don't have a UI but they can create a status bar notification.
    - **Content providers**: A content provider manages app data that you can store in a storage location that the app can access. The content provider allows other apps to query and/or modify the data if the content provider allows it.
5. AndroidManifest.xml is a required file for every android app. The file itself contains essential information about your app which it describes to the Android build tools, the Android OS, and Google Play.
6. Resources are just additional files and static content that your code might use.