Baby Objects

In Java, you can make your own variable types.

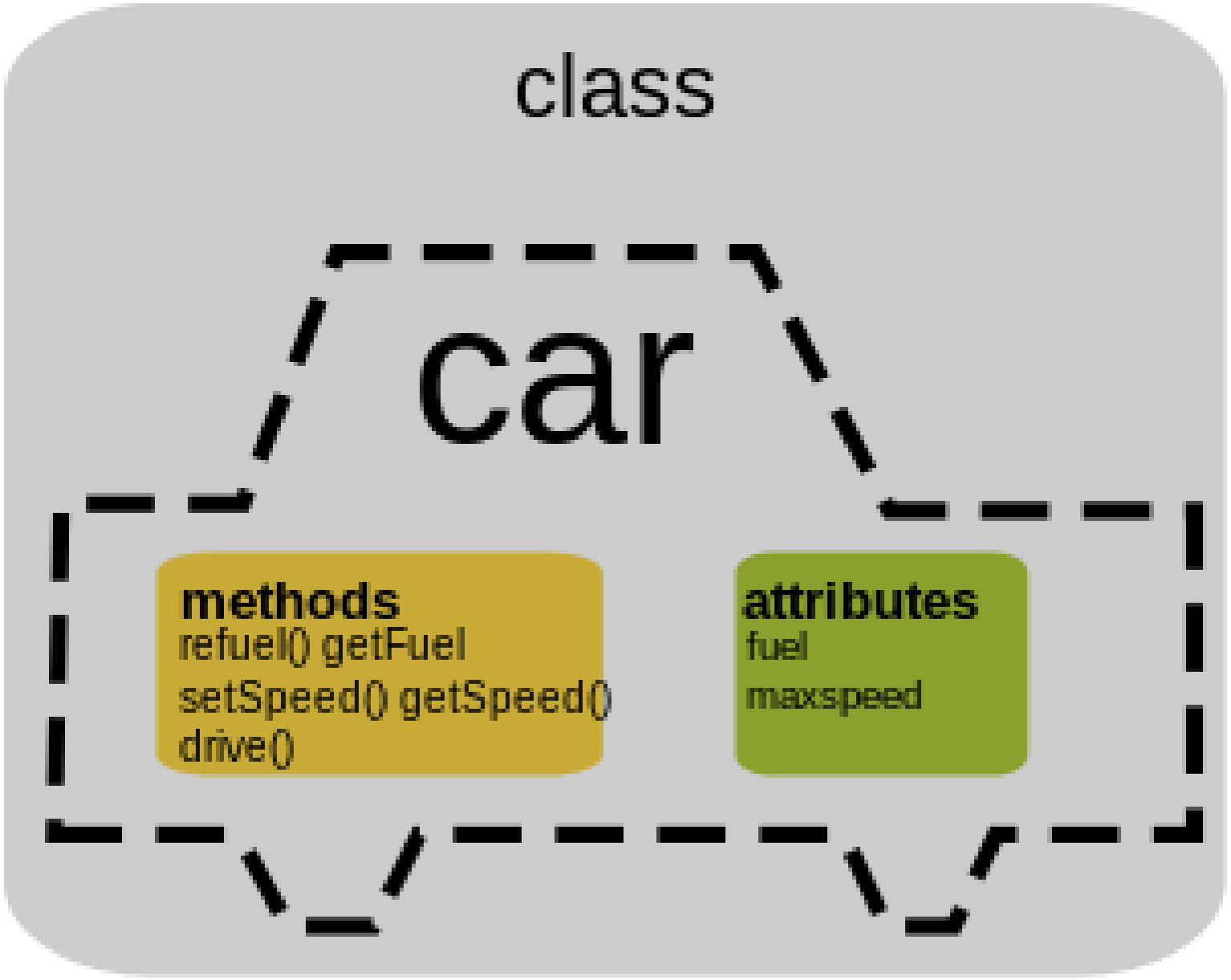
These are called objects.

Instead of simply storing one variable, it can group a number of variables together into one unit.

Objects:

- A bottle – size, name, recipe, price
- A basket – Holds other objects
- A baked good – name, size, price

Complex details grouped together.

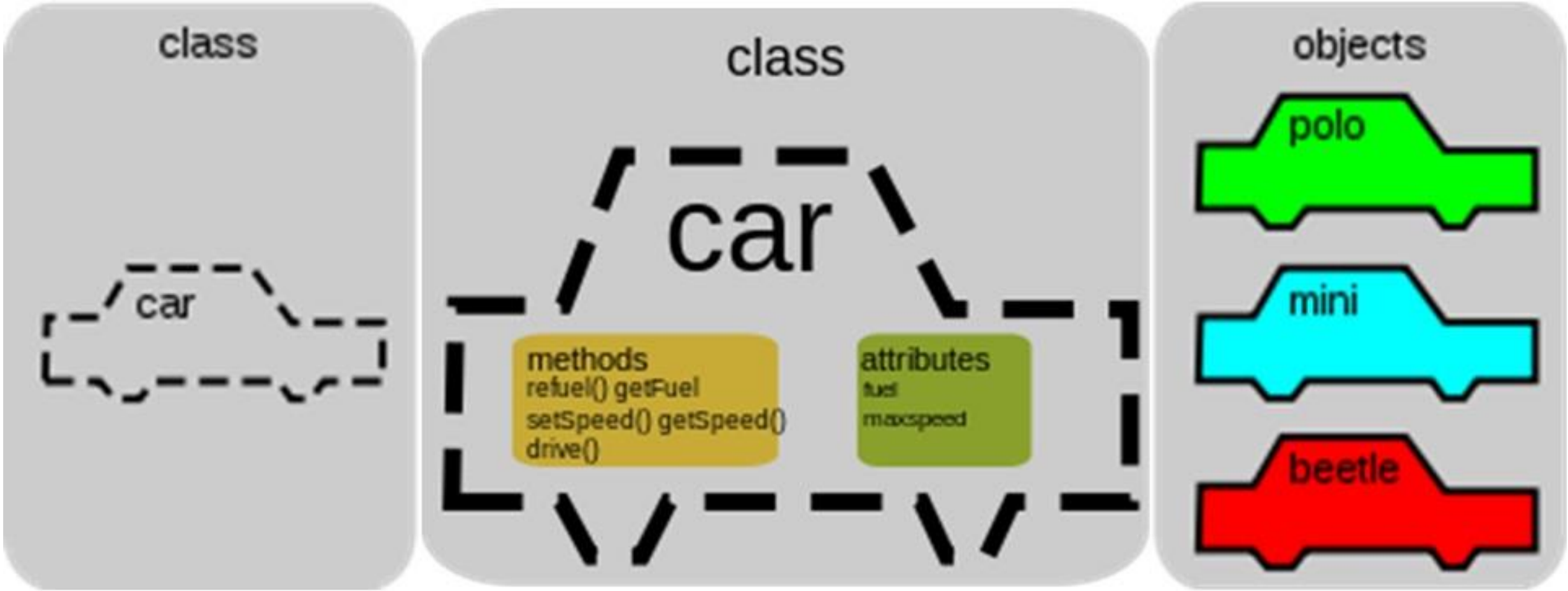


Objects

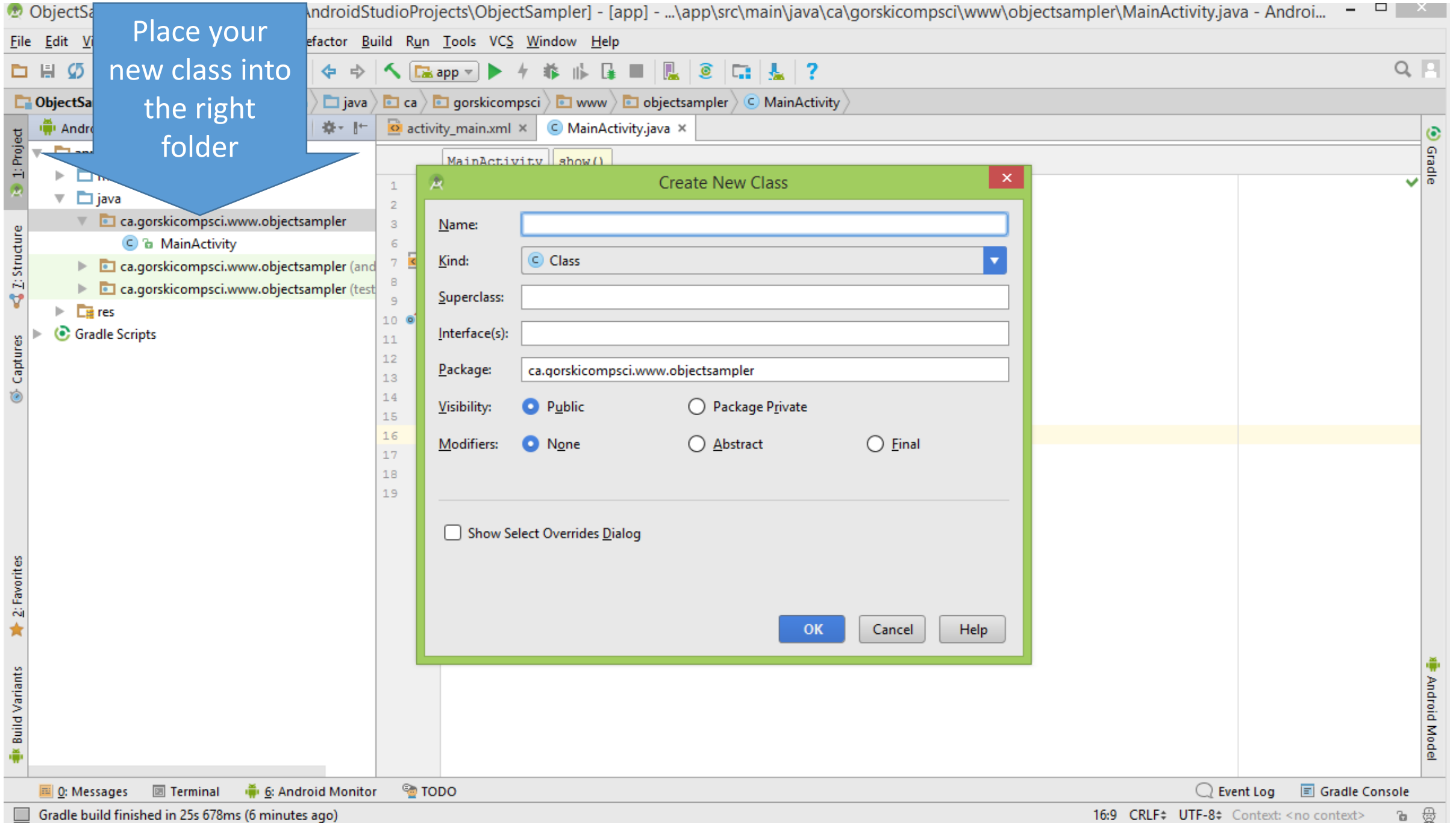
Have two pieces:

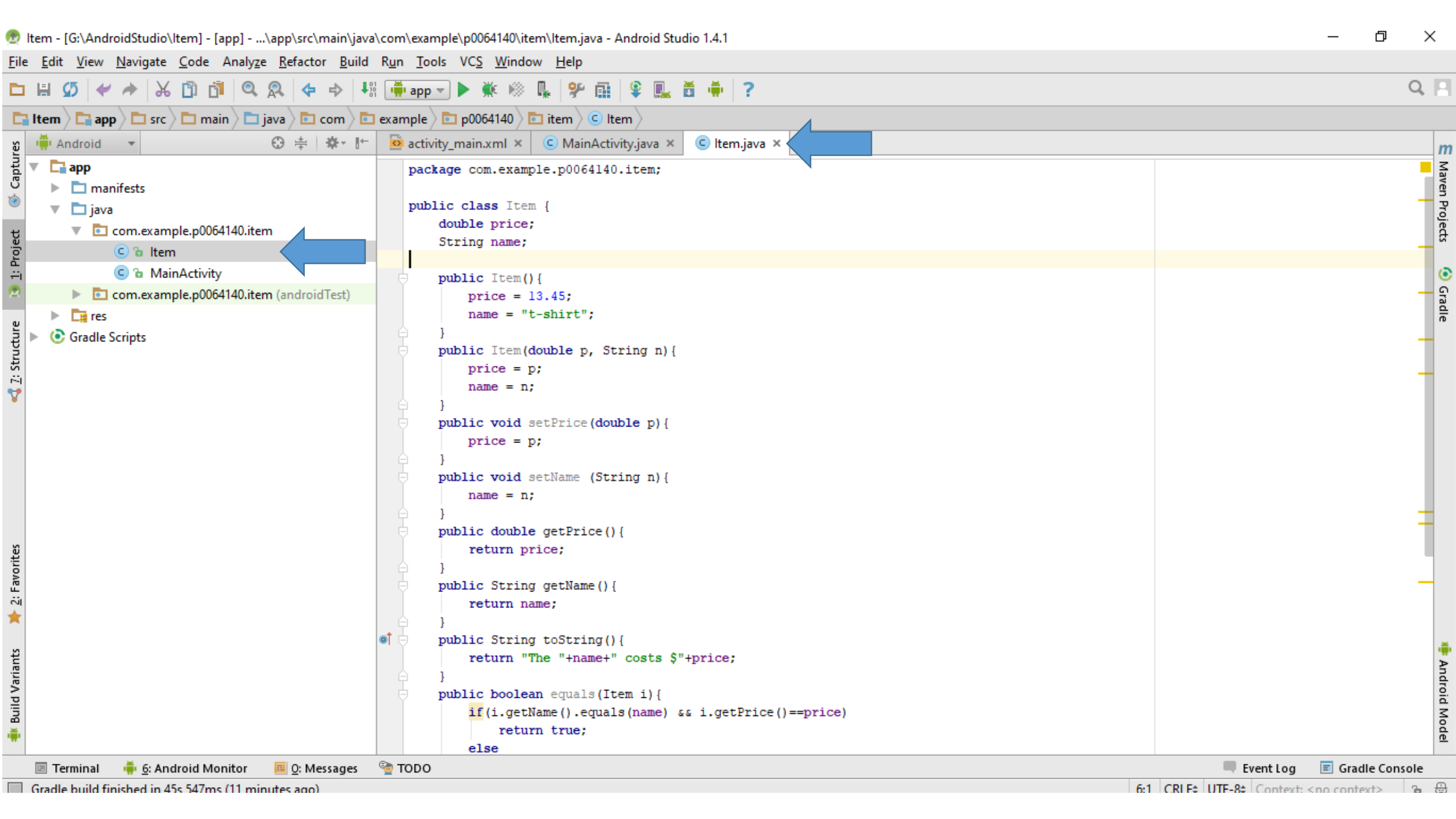
1. Data – variables
2. Methods that use those variables.

These pieces are grouped into a class.



A class is like a template for a brand new variable type.
You can make many variables using a class as the variable type.


Place your new class into the right folder
Create New Class
Name:
Kind:
Class
Superclass:
Interface(s):
Package:
ca.gorskicompsci.www.objectsampler
Visibility:
Public
Package Private
Modifiers:
None
Abstract
Final
Show Select Overrides Dialog
OK
Cancel
Help


Item - [G:\AndroidStudio\Item] - [app] - ...\app\src\main\java\com\example\p0064140\item\Item.java - Android Studio 1.4.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
Item app src main java com example p0064140 item Item
Android
app
manifests
java
com.example.p0064140.item
Item
MainActivity
com.example.p0064140.item (androidTest)
res
Gradle Scripts
activity_main.xml
MainActivity.java
Item.java
package com.example.p0064140.item;
public class Item {
double price;
String name;
public Item(){
price = 13.45;
name = "t-shirt";
}
public Item(double p, String n){
price = p;
name = n;
}
public void setPrice(double p){
price = p;
}
public void setName (String n){
name = n;
}
public double getPrice(){
return price;
}
public String getName(){
return name;
}
public String toString(){
return "The "+name+" costs $"+price;
}
public boolean equals(Item i){
if(i.getName().equals(name) && i.getPrice()==price)
return true;
else
Captures
1: Project
7: Structure
2: Favorites
Build Variants
Maven Projects
Gradle
Android Model
Terminal
6: Android Monitor
0: Messages
TODO
Event Log
Gradle Console
Gradle build finished in 45s 547ms (11 minutes ago)
6:1
CRLF
UTF-8
Context: <no context>

```
public class Item {
    double price;
    String name;
```

```
public Item(){
  price = 13.45;
  name = "t-shirt";
}
public Item(double p, String n){
  price = p;
  name = n;
}
```

```
public double getPrice(){
  return price;
}
public String getName(){
  return name;
}
public String toString(){
  return "The "+name+" costs $"+price;
}
```

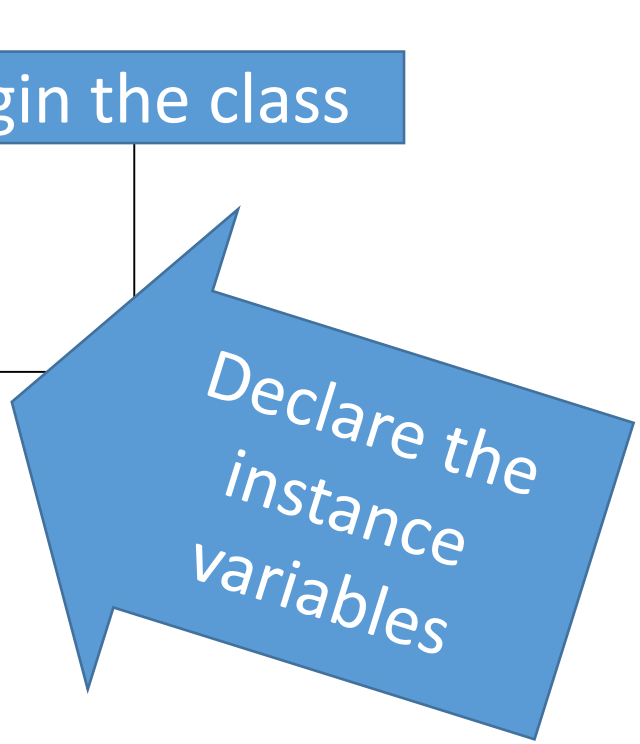
```
public void setPrice(double p){
  price = p;
}
public void setName (String n){
  name = n;
}
```

```
public boolean equals(Item i){
  if(i.getName().equals(name)
          && i.getPrice()==price)
     return true;
  else
     return false;
}
public int compareTo(Item i){
  //on the basis of price
  if(i.getPrice()>price)
      return -1;
  else if (i.getPrice()==price)
      return 0;
  else
      return 1;
}}
```

Instance Variables:

- The variables that you want to store for your object.
- Your object will group these variables together into a new complex type

```
public class Item {
    private double price;
    private String name;
```



Constructors

- Initialize and set up memory.
- You need a default AND one with parameters for each instance variable.

```
public Item(){
   price = 13.45;
   name = "t-shirt";
}

public Item(double p, String n){
   price = p;
   name = n;
}
```

Default.
Put a value in each instance variable.

Parameter for each instance variable.

Take each parameter, assign to instance variable.

```
public Item(){
   price = 13.45;
   name = "t-shirt";
}
```

Constructors are special.

- They have no return type because the type they return is themselves (in this case, an Item).
- They must have the same name as the class.
- When they are called, they are called with the word new and the class name.

```
Item shoe = new Item(23.45, "flip-flops");
Item shirt = new Item();
```

Mutators

- Change memory
- You need one for each instance variable.

```
public void setPrice(double p){
   price = p;
}
public void setName (String n){
   name = n;
}
```

Parameter for the instance variable.

Take parameter, assign to the right instance variable.

Accessors

- Access what is stored in memory.
- You need one for each instance variable.

```
public double getPrice(){
   return price;
}
public String getName(){
   return name;
}
public String toString(){
   return "The "+name+" costs $"+price;
}
```

Return type matches the instance variable type.

Return correct instance variable

Make a sentence out of the variables.

Facilitator: Equals

- Sees if two of your new type are equal

Pass in an object that is the same type as your class

```
public boolean equals(Item i){
   if(i.getName().equals(name)
            && i.getPrice()==price)
      return true;
   else
      return false;
}
```

For each instance variable, see if it matches the parameters' value

Return true if all instance variables match, false otherwise.

Facilitator: CompareTo

- Sees how two of your new type compare, for sorting

```
public int compareTo(Item i){
   //on the basis of price
   if(i.getPrice()>price)
        return -1;
   else if (i.getPrice()==price)
        return 0;
   else
        return 1;
}}
```

If the parameter is bigger, return -1

If they are the same, return 0

If the parameter is smaller, return 1

```
public class Item {
    double price;
    String name;
```

```
public Item(){
  price = 13.45;
  name = "t-shirt";
}
public Item(double p, String n){
  price = p;
  name = n;
}
```

```
public double getPrice(){
  return price;
}
public String getName(){
  return name;
}
public String toString(){
  return "The "+name+" costs $"+price;
}
```

```
public void setPrice(double p){
  price = p;
}
public void setName (String n){
  name = n;
}
```

```
public boolean equals(Item i){
  if(i.getName().equals(name)
          && i.getPrice()==price)
     return true;
  else
     return false;
}
public int compareTo(Item i){
  //on the basis of price
  if(i.getPrice()>price)
      return -1;
  else if (i.getPrice()==price)
      return 0;
  else
      return 1;
}}
```

Item - [G:\AndroidStudio\Item] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 1.4.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Item > app > src > main > res > layout > activity_main.xml

Android

app
- manifests
- java
  - com.example.p0064140.item
    - Item
    - MainActivity
  - com.example.p0064140.item (androidTest)
- res
- Gradle Scripts

activity_main.xml | MainActivity.java | Item.java

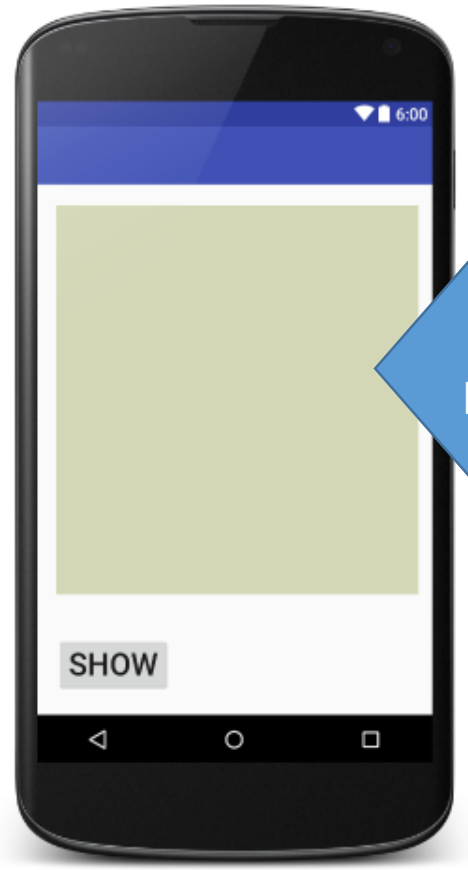
```
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_horizontal"
    android:orientation="vertical">

    <TextView
        android:id="@+id/TextArea"
        android:padding="10dp"
        android:textSize="30dp"
        android:layout_margin="20dp"
        android:layout_width="350dp"
        android:layout_height="wrap_content"
        android:background="#d3d7b6"
        android:inputType="textMultiLine"
        android:maxLines="10"
        android:minLines="10"
        android:scrollbars="vertical" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:onClick="show"
        android:textSize="30dp"
        android:text="show" />

</LinearLayout>
```

Multiline TextView

Design | Text

Preview

Nexus 4 | AppTheme | MainActivity | 23



Looks like this

Terminal | 6: Android Monitor | 0: Messages | TODO | Event Log | Gradle Console

```java
public class Item {
    double price;
    String name;

    public Item(){
        price = 13.45;
        name = "t-shirt";
    }
    public Item(double p, String n){
        price = p;
        name = n;
    }
    public void setPrice(double p){
        price = p;
    }
    public void setName (String n){
        name = n;
    }
    public double getPrice(){
        return price;
    }
    public String getName(){
        return name;
    }
    public String toString(){
        return "The "+name+" costs $"+price;
    }
    public boolean equals(Item i){
        if(i.getName().equals(name)
                && i.getPrice()==price)
            return true;
        else
            return false;
    }
    public int compareTo(Item i){
        //on the basis of price
        if(i.getPrice()>price)
            return -1;
        else if (i.getPrice()==price)
            return 0;
        else
            return 1;     }
}
```

```
public void show(View view) {
    TextView textArea = (TextView) findViewById(R.id.TextArea);
    Item shoe = new Item(23.45, "flip-flops");
    Item shirt = new Item();
    textArea.append(""+shoe.toString ());
    textArea.append("\n"+shirt.toString ());
    textArea.append("\n"+shoe.getPrice ());
    shirt.setPrice (100.98);
    textArea.append("\n"+shirt.toString ());
    textArea.append("\n" + shirt.compareTo(shoe));
    textArea.append("\n" + shoe.equals(shirt));
}
```

```
public class Item {
    double price;
    String name;
```

```
public Item(){
  price = 13.45;
  name = "t-shirt";
}
public Item(double p, String n){
  price = p;
  name = n;
}
```

```
public double getPrice(){
  return price;
}
public String getName(){
  return name;
}
public String toString(){
  return "The "+name+" costs $"+price;
}
```

```
public void setPrice(double p){
  price = p;
}
public void setName (String n){
  name = n;
}
```

```
public boolean equals(Item i){
  if(i.getName().equals(name)
        && i.getPrice()==price)
     return true;
  else
     return false;
}
public int compareTo(Item i){
  //on the basis of price
  if(i.getPrice()>price)
      return -1;
  else if (i.getPrice()==price)
      return 0;
  else
      return 1;
}}
```

ObjectSampler - [C:\Users\AJGorski\AndroidStudioProjects\ObjectSampler] - [app] - ...\app\src\main\java\ca\gorskicompsci\www\objectsampler\MainActivity.java - Androi...

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

ObjectSampler > app > src > main > java > ca > gorskicompsci > www > objectsampler > MainActivity

activity_main.xml | MainActivity.java | Animal.java

Activity > onCreate()

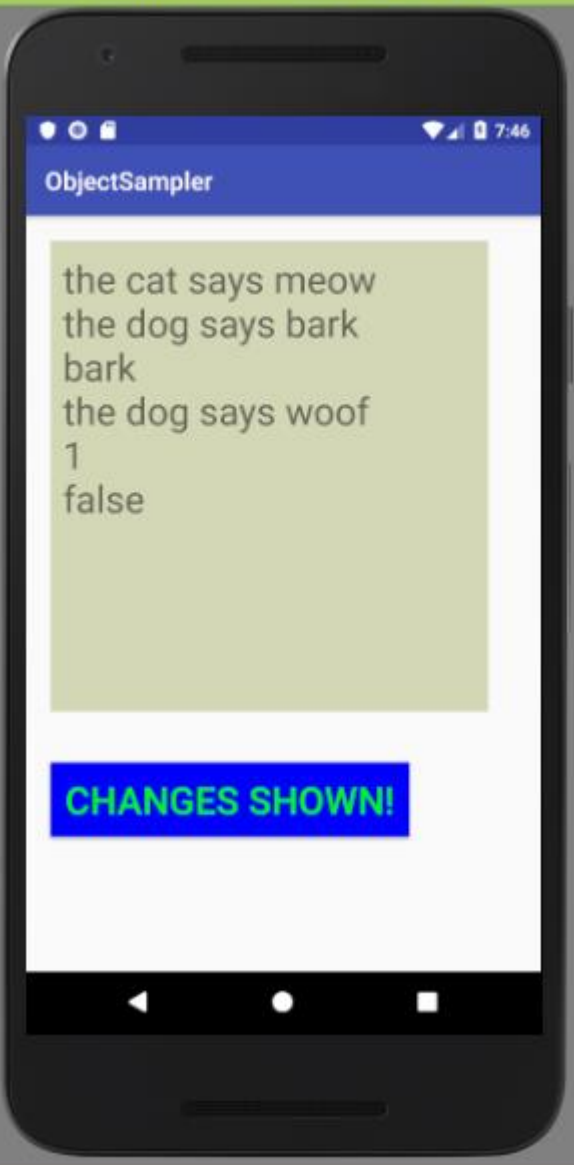
```
ic class MainActivity extends AppCompatActivity {

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);


}


public void show(View view) {
    Button b = (Button) findViewById(R.id.show);
    b.setText("Changes Shown!");
    b.setBackgroundColor(Color.BLUE);
    b.setTextColor(Color.GREEN);

    TextView textArea = (TextView) findViewById(R.id.TextArea);
    Animal spot = new Animal ();
    textArea.append(""+spot.toString ());
    Animal fluffy = new Animal ("dog", "bark");
    textArea.append("\n"+fluffy.toString ());
    textArea.append("\n"+fluffy.getSound ());
    fluffy.setSound ("woof");
    textArea.append("\n"+fluffy.toString ());
    textArea.append("\n"+spot.compareTo (fluffy));
    textArea.append("\n"+spot.equals (fluffy));
}
```

activity_main.xml | MainActivity.java | Animal.java

Animal

```
package ca.gorskicompsci.www.objectsampler;

public class Animal {

    private String noise;
    private String type;

    public Animal () {
        noise = "meow";
        type = "cat";
    }

    public Animal (String a, String s) {
        noise = s;
        type = a;
    }

    public String toString () {
        return "the " + type + " says " + noise;
    }

    public String getSound () {
        return noise;
    }

    public String getAnimal () {
        return type;
    }
```

Android Emulator - Nexus_5X_API_26:5554



0: Messages | Terminal | 6: Android Monitor | 4: Run | TODO | Event Log | Gradle Console