Summer 2, 2019 - CS 4520/CS5520 – Mobile Application Development

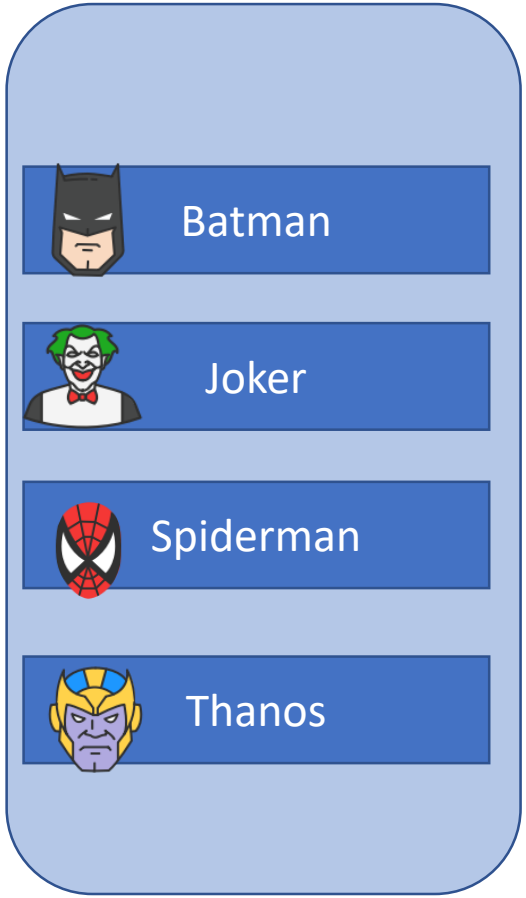
Instructor: Pratheep Kumar Paranthaman, Ph.D.,

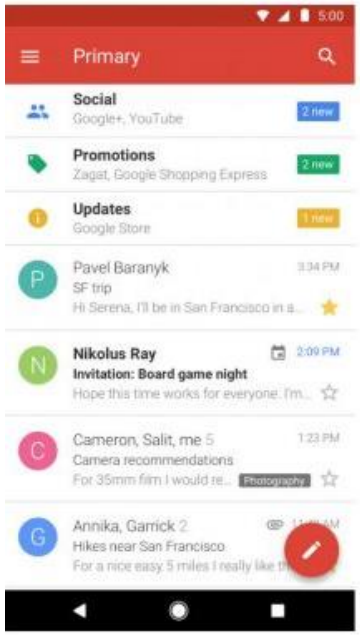
Playing an Audio file

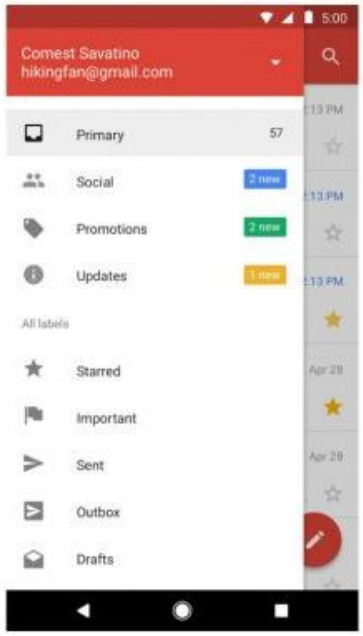
```
//Create mediaplayer object
MediaPlayer mediaPlayer = new MediaPlayer();

//Assign the audio resource

mediaPlayer = MediaPlayer.create(this, audioResID);


//Start the player

mediaPlayer.start();
```

Custom Adapters

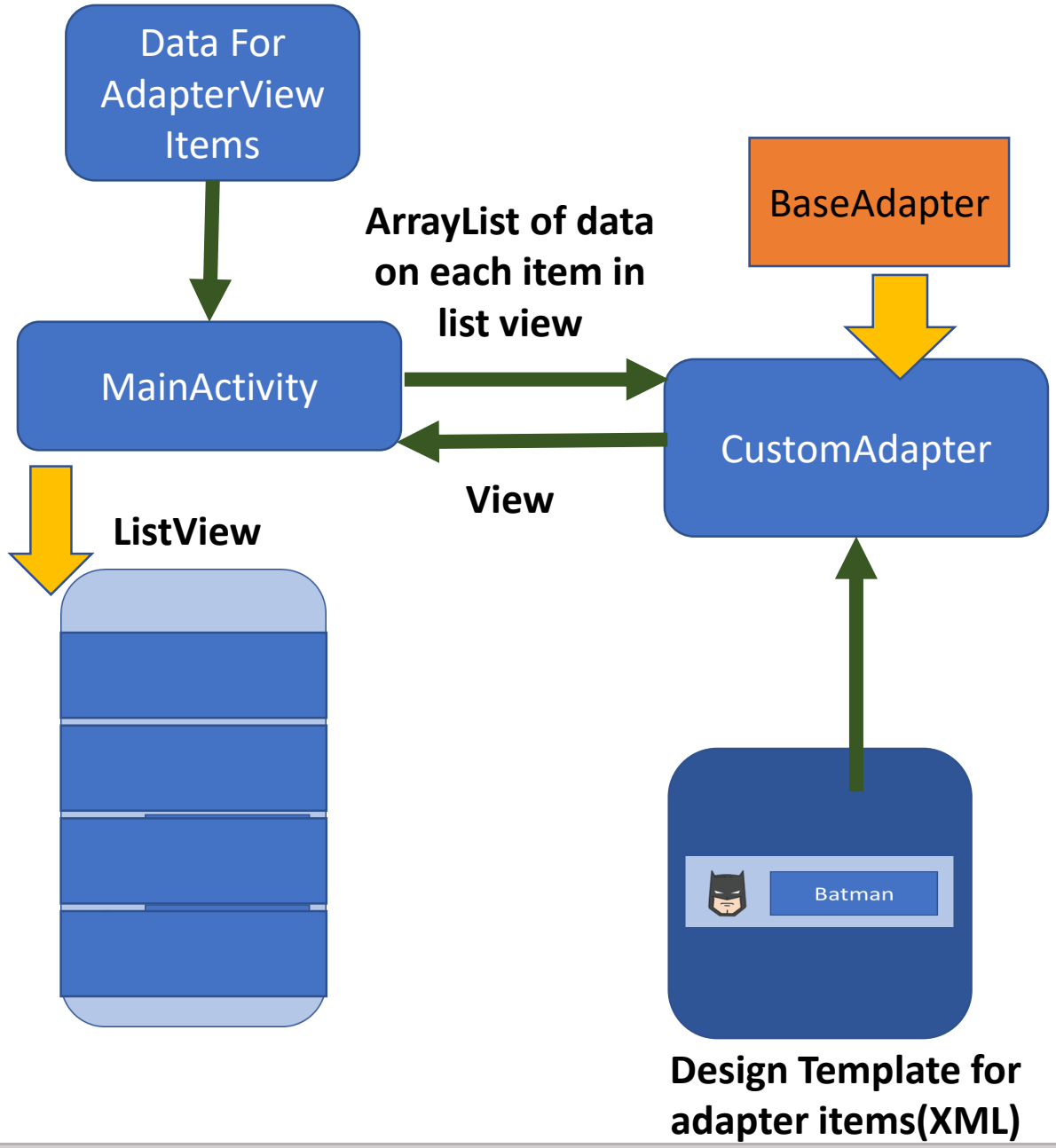
Implementation of Custom Adapter







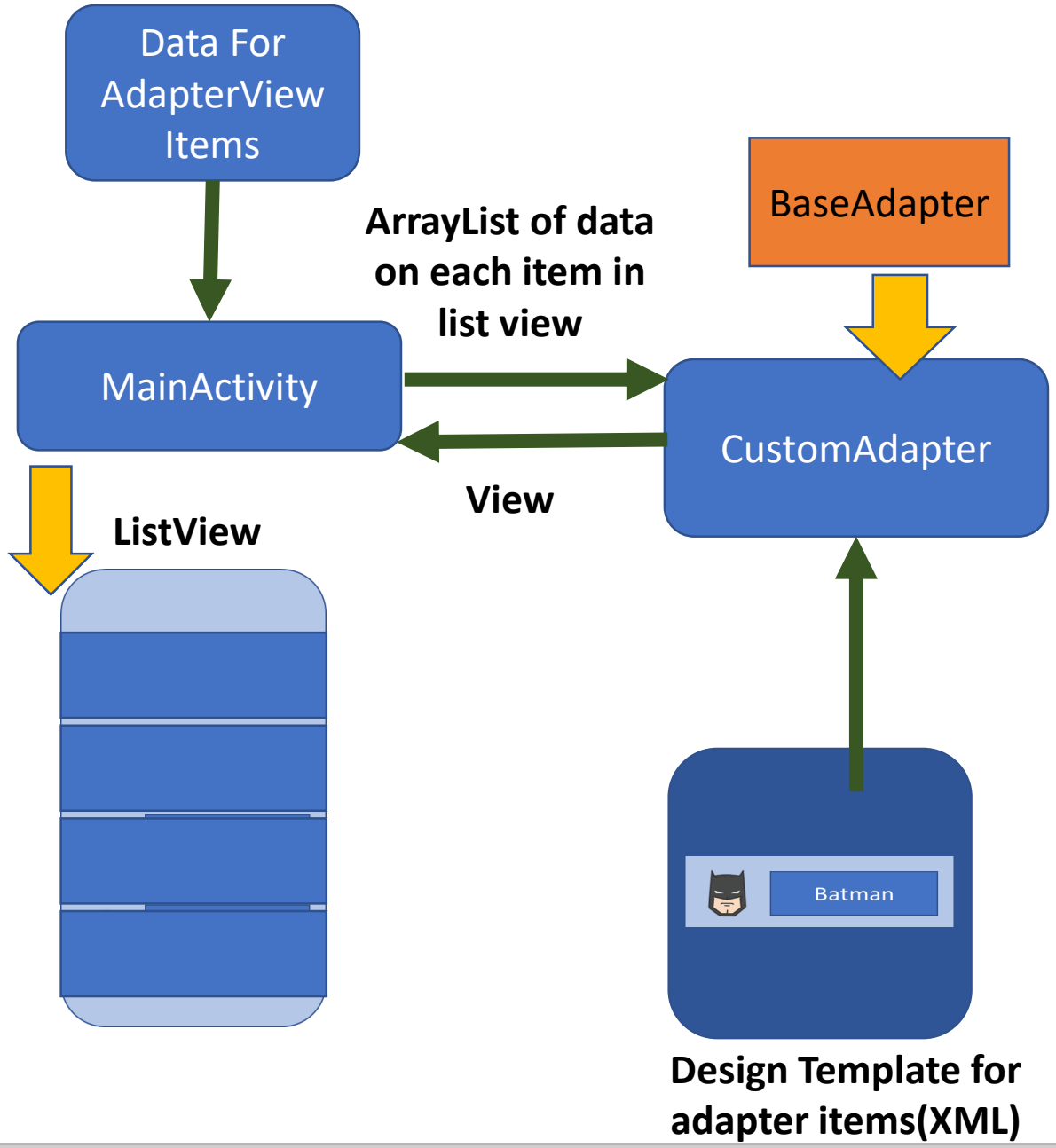
[thenextweb.com/apps/2018/02/16/gmail-go-is-almost-identical-to-the-original-android-app-but-half-the-size/]

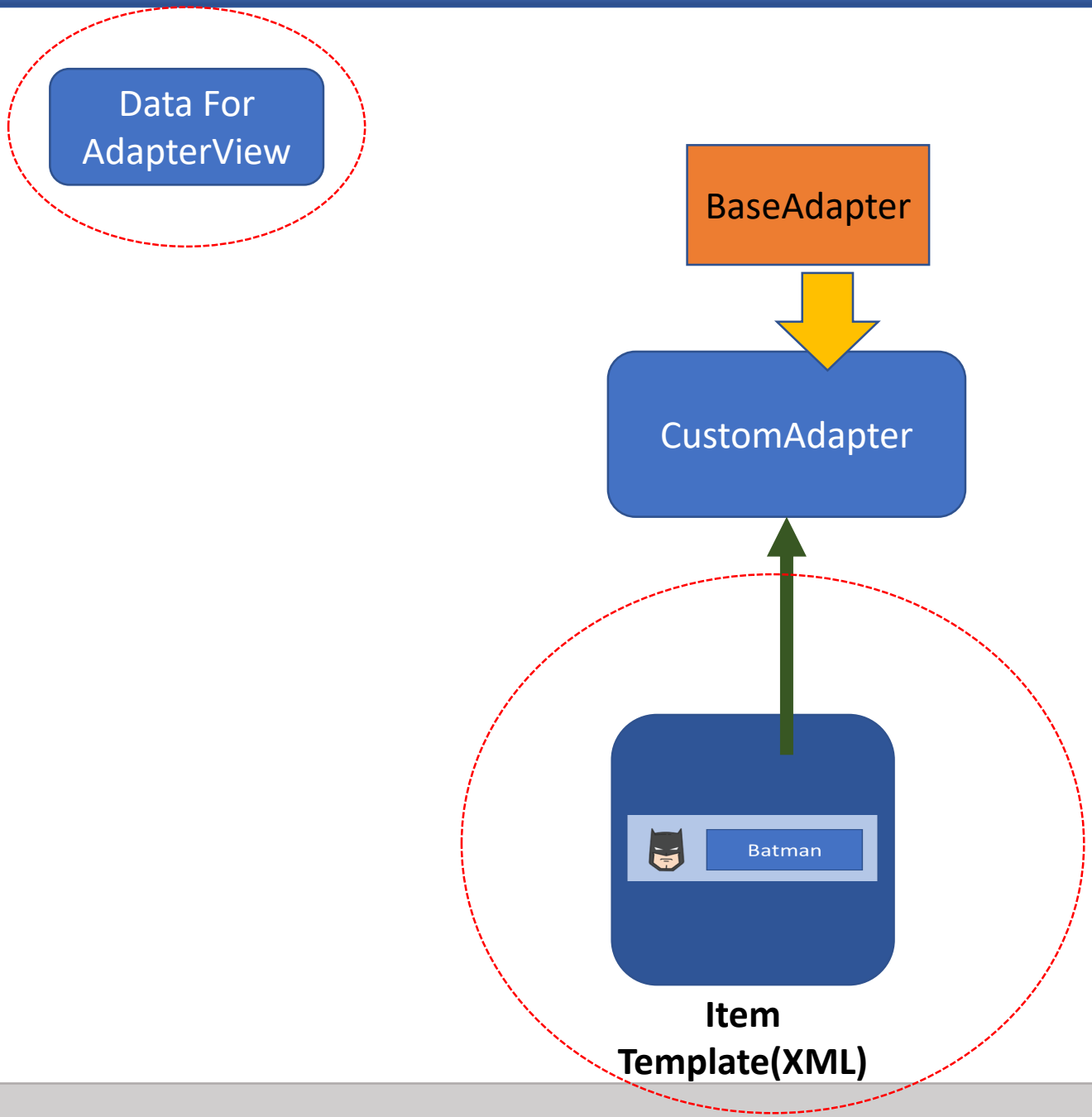

Data For AdapterView Items
ArrayList of data on each item in list view
BaseAdapter
MainActivity
CustomAdapter
View
ListView
Batman
Design Template for adapter items(XML)

Steps for creating Custom Adapter

- Step 1 - Configuring the AdapterView Item
- Step 2 - Creating CustomAdapter class
- Step 3 - Implementing the BaseAdapter methods

- **Using your CustomAdapter in the MainActivity**
  - **Create a customAdapter**
  - **Set the CustomerAdapter to your ListView**
  - **Set clickListener to your listview**


Data For AdapterView Items
ArrayList of data on each item in list view
BaseAdapter
MainActivity
CustomAdapter
View
ListView
Batman
Design Template for adapter items(XML)

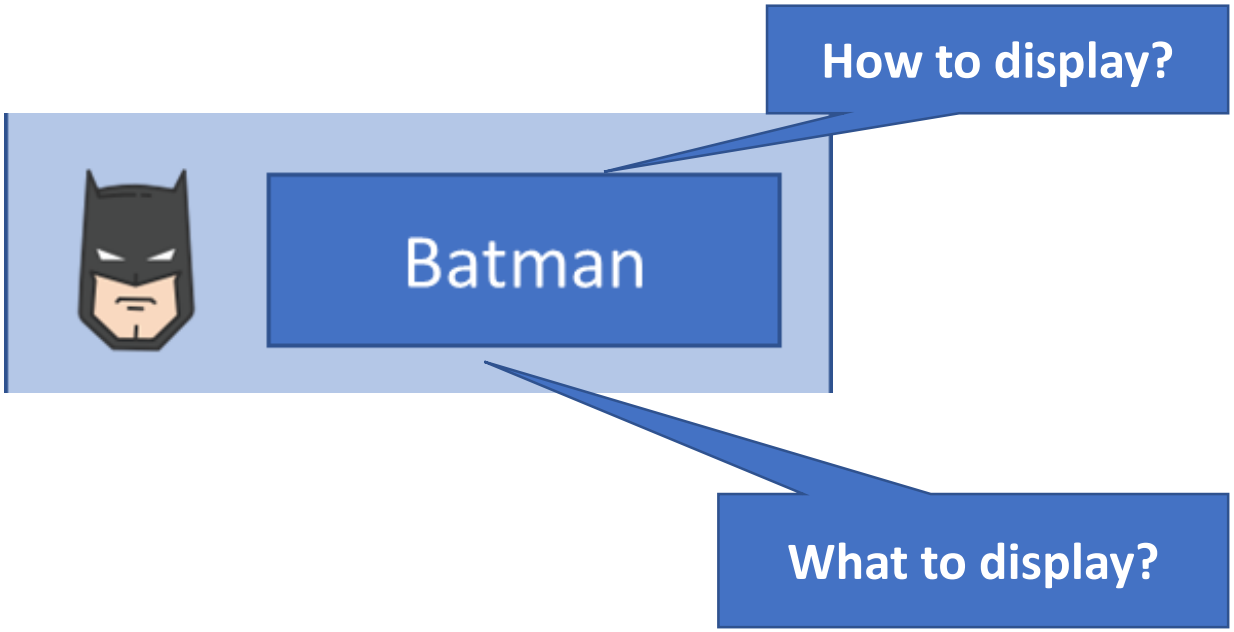

Data For AdapterView
BaseAdapter
CustomAdapter
Batman
Item Template(XML)

Step 1 – Configuring the AdapterView Item

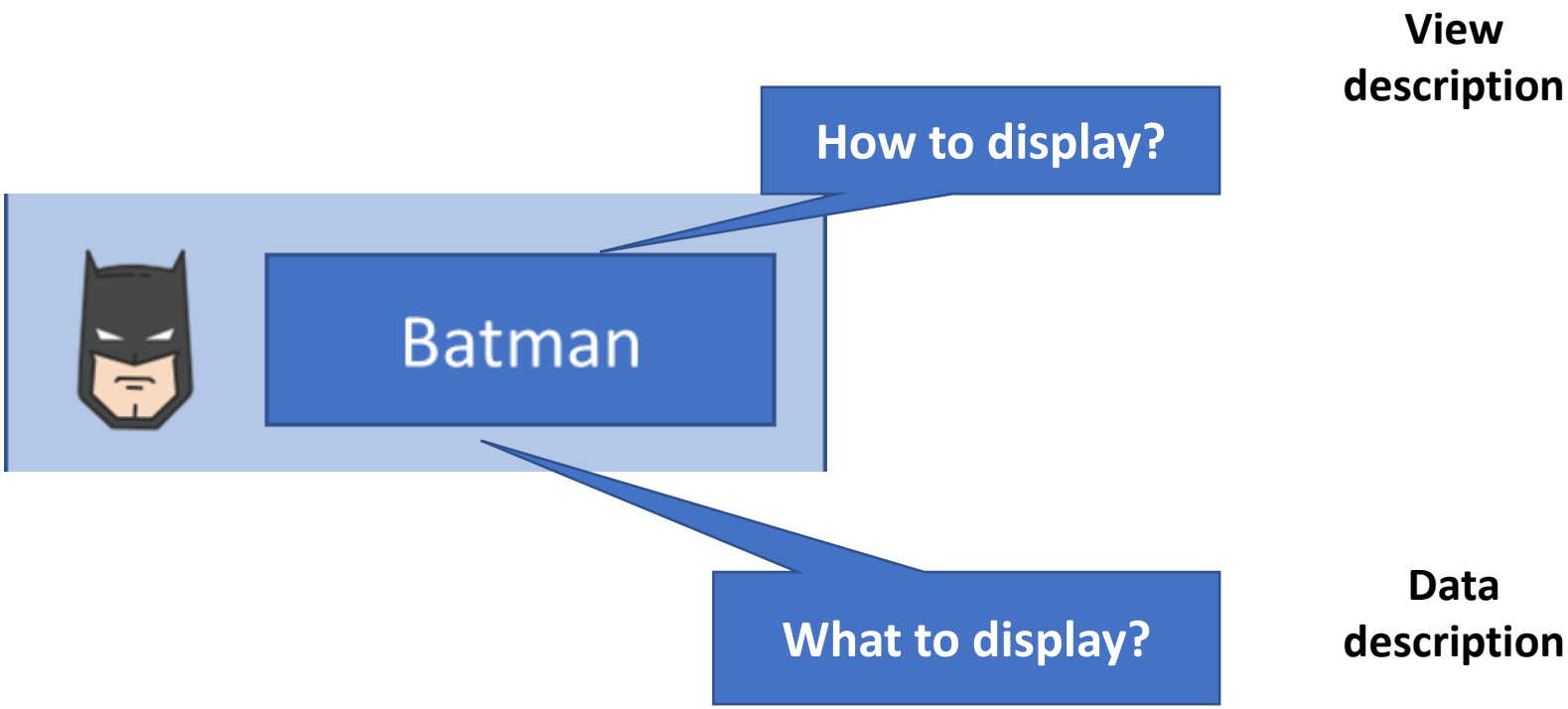
Configuring the AdapterView Item

You need to author the looks of each item in your CustomAdapter

Configuring the AdapterView Item

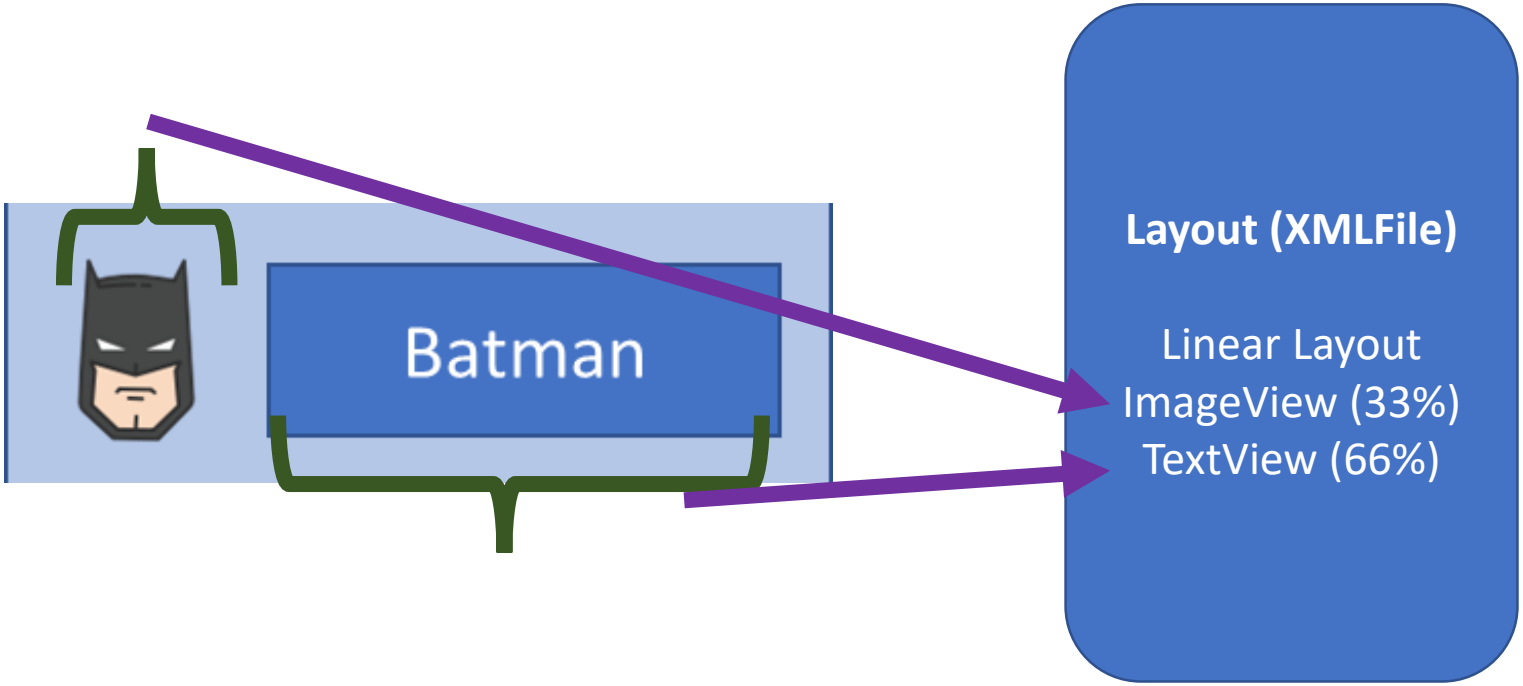


Configuring the AdapterView Item



Adapter Item View Description – XML file

- How should it look?

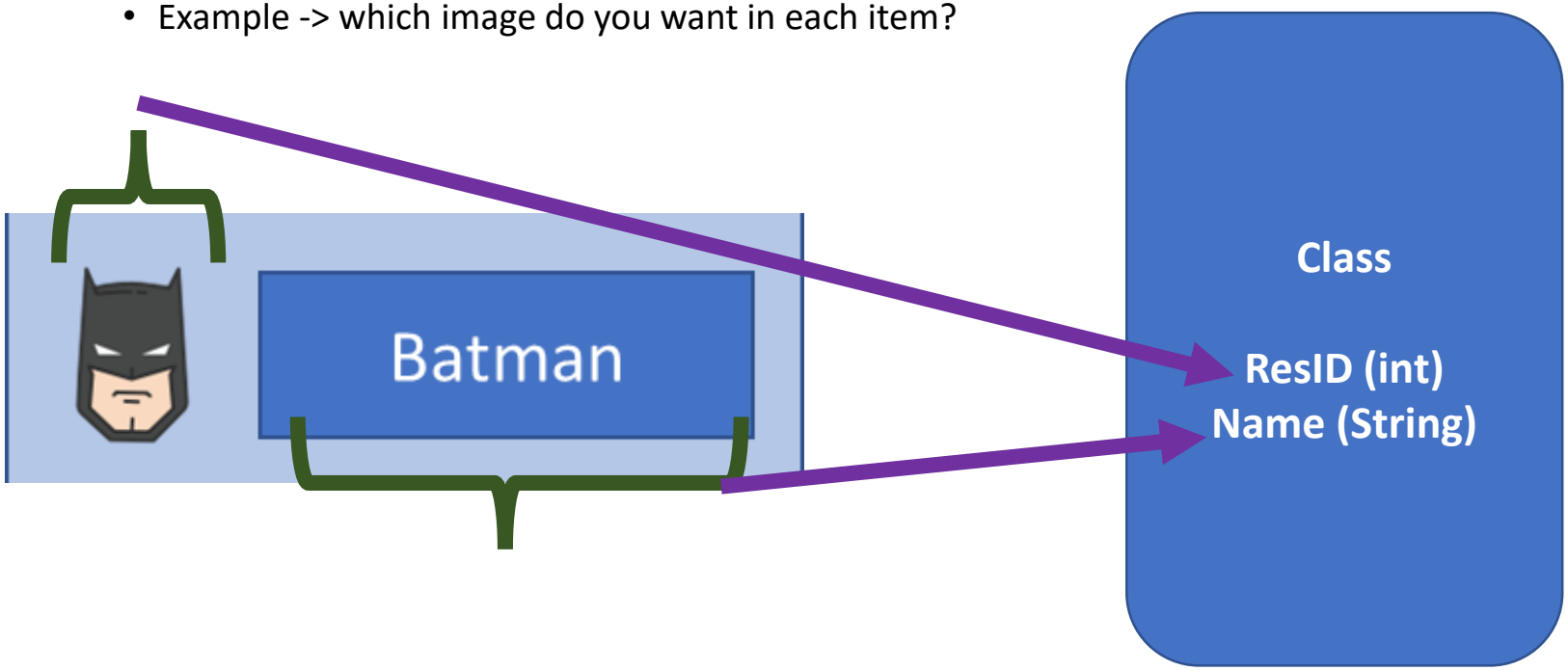


Adapter Item View Description – XML file

- Create a new layout file in the Layout directory
  - Layout -> new -> Layout resource file -> character_info_adapter_item

Data in Adapter Item – Java Class

- What should it contain?
  - custom data structure to hold the data points that you want to include in the Adapter Item
    - Example -> which image do you want in each item?



Custom Adapter Configuration

- ***FavoriteCharacterAdapterItem*** Class
  - Name
  - Image

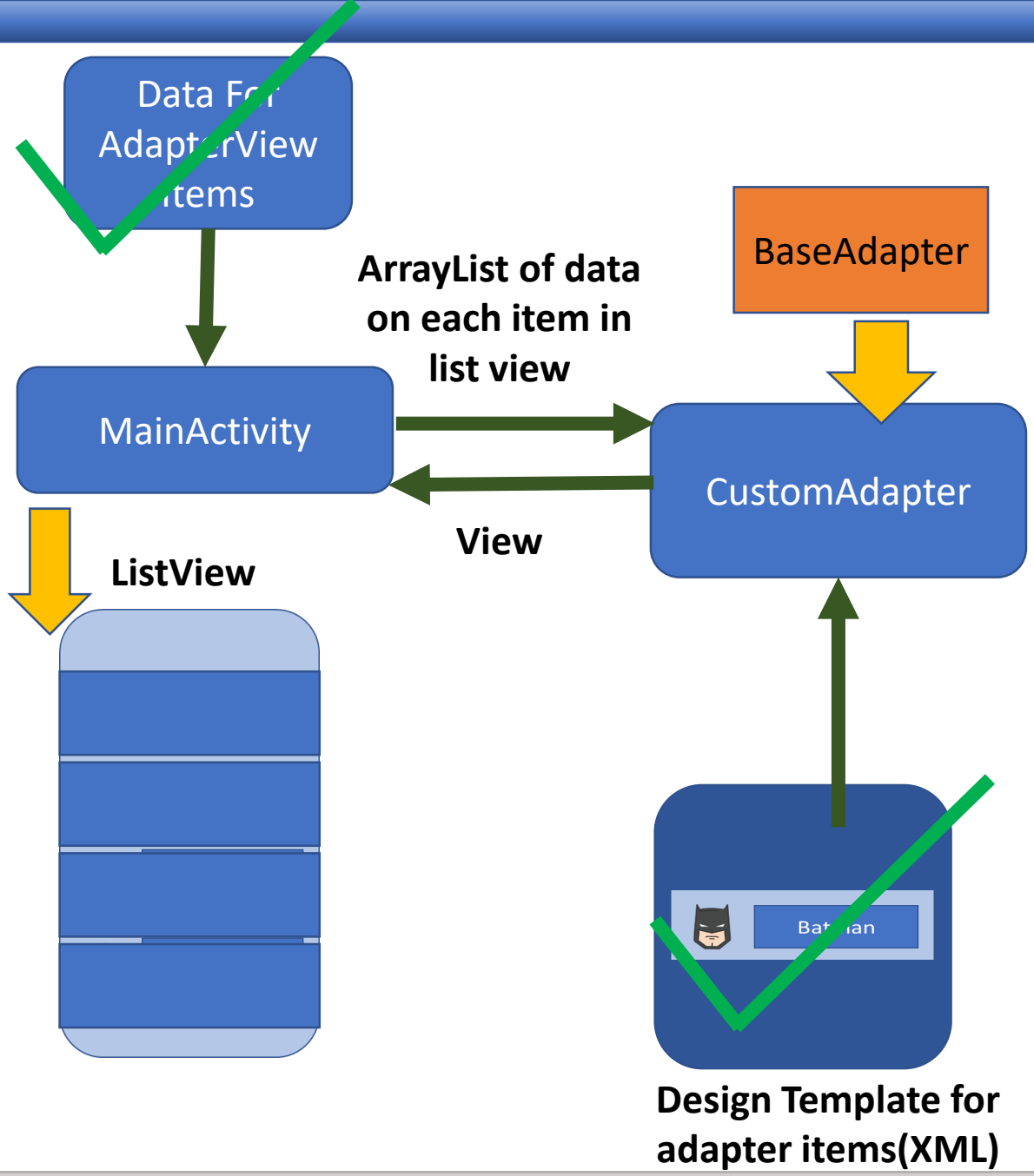
- DataTypes
  - Name -> String
  - Image -> int

Creating CustomAdapter class

- Create a Package called **“utils”** in your project
- Include Java class named **” FavoriteCharacterAdapterItem**” in the utils
- Let the **FavoriteCharacterAdapterItem** extend from base adapter

Creating class for AdapterView Data points

- **Step 1:** Create a custom class(FavoriteCharacterAdaterItem) to store the details data points in each custom Item on the listView
  - Create a package and name it as “utils” -> set the path to be under main
  - Create a class file under this utils folder
  - Declare the necessary elements in the class
    - Variables to hold the information about each character
    - Getters and Constructor


Data For AdapterView Items
ArrayList of data on each item in list view
BaseAdapter
MainActivity
CustomAdapter
View
ListView
Design Template for adapter items(XML)

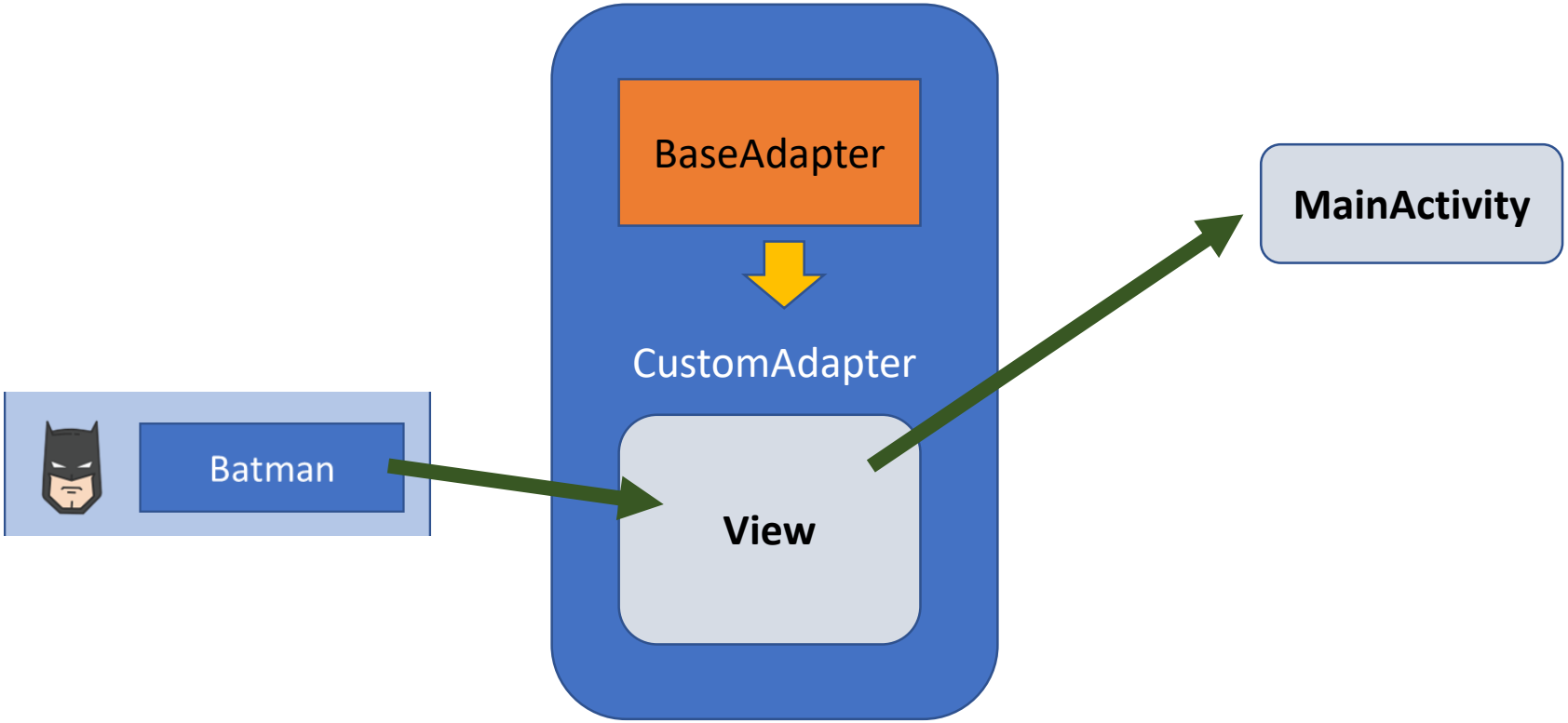
Step 2 - Creating CustomAdapter class

Step 2 - Creating CustomAdapter class

- Create a CustomerAdapter class
- Extend Base Adapter
- Implement the methods of Base Adapter
- Create a Constructer to receive data from MainActivity
- **Create View (based on the XML definition and data points)**

return View

Creating CustomAdapter class



Let’s create the custom adapter class

Creating CustomAdapter class

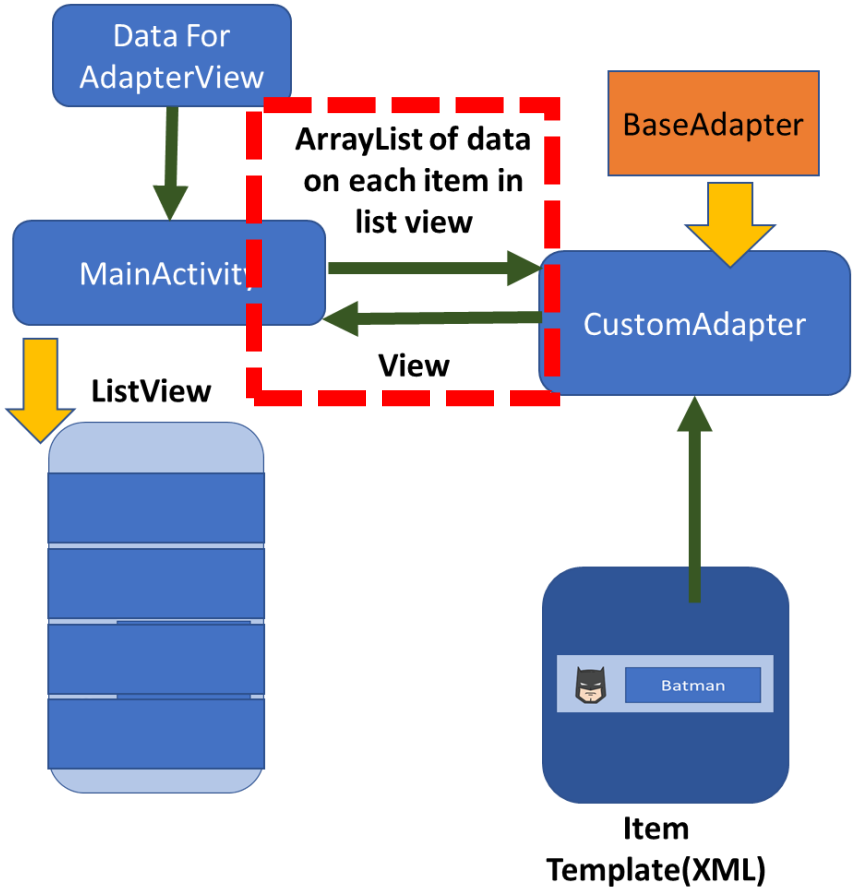
- Create a **CustomCharacterAdapter** java class and put in under "**Utils**" folder

- Extend the BaseAdapter

BaseAdapter

- **getCount()- total number of items to be displayed on the listview**
- getItem(int itemIndex) – extracts the data of the item at specific location in the list
- getItemId(int itemId) – extracts the item id on the adapter; this returns long
- **getView(int i, View view, ViewGroup viewGroup) –returns the view of items in the list view**
  - **Here's where you will specify how each item should look like**

Create a Constructor to receive data from MainActivity

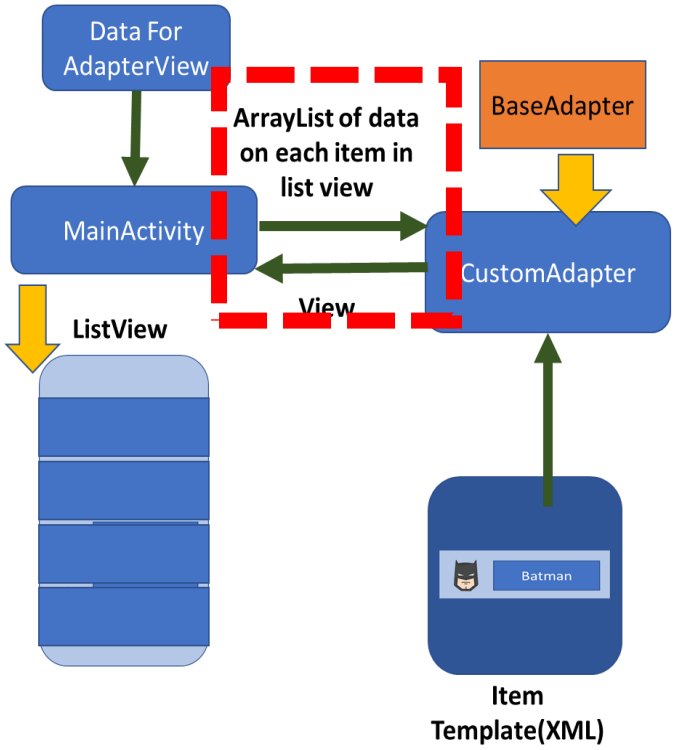


Create a Constructor to receive data from MainActivity

1. Context – of the MainActivity

2. ArrayList - of type **FavoriteCharacterAdapterItem**

ImageResId	CharacterName
R.Drawable.batman	“Batman”
R.Drawable.joker	“Joker”
R.Drawable.spiderman	“Spiderman”
R.Drawable.thanos	“Thanos”



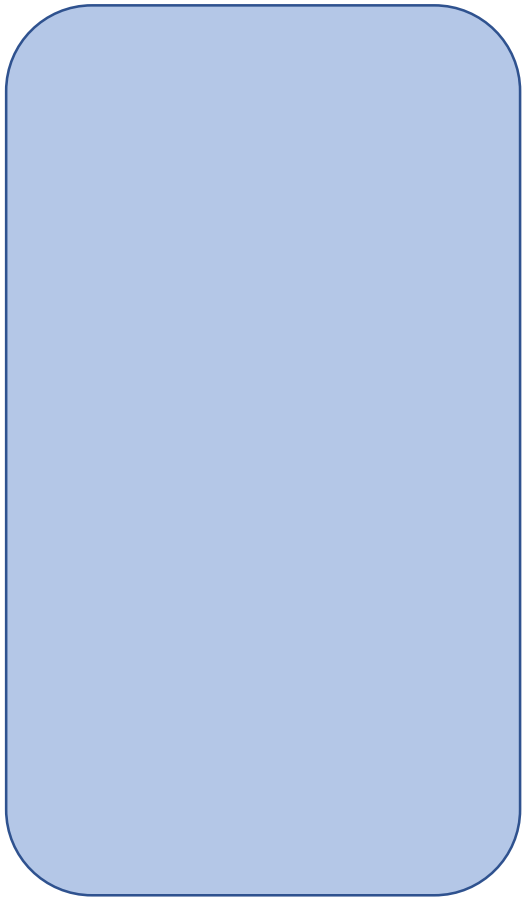
Step 2 - Creating CustomAdapter class

- ~~Create a CustomerAdapter(Java class)~~
- ~~Extend Base Adapter~~
- ~~Implement the methods of Base Adapter~~
- ~~Create a Constructer to receive data from MainActivity~~
- **Create View (based on the XML definition and data points)**

return View

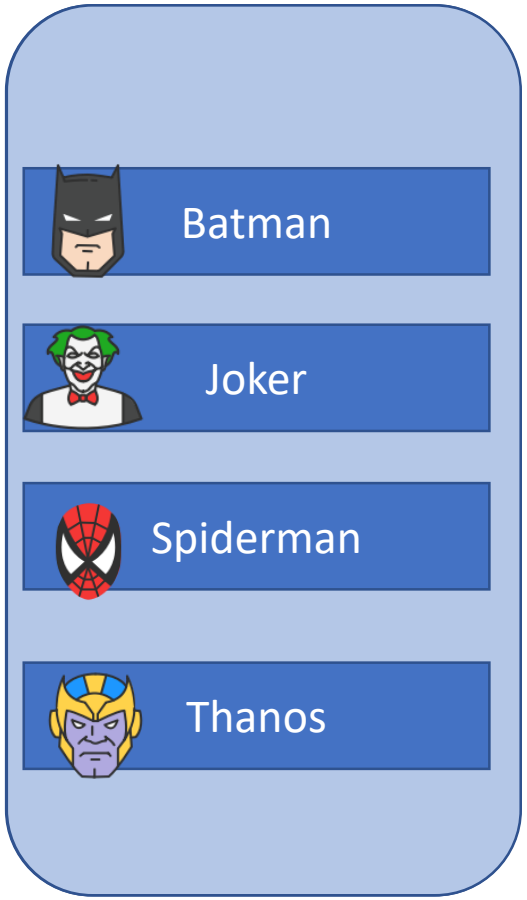
Creating and returning the View

- Determine how many items should be generated
  - We can use **getCount**() to extract how items should be shown in the ListView -> ***in our case it's going to be 4***



Creating and returning the View

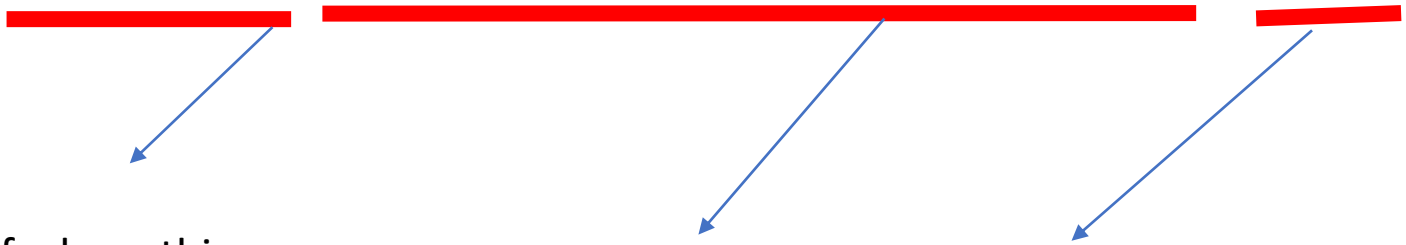
- Determine how many items should be generated
  - We can use **getCount**() to extract how items should be shown in the ListView -> ***in our case it's going to be 4***



Creating and returning the View

Use LayoutInflator to load Root View (the XML Template file) during runtime

```
View.inflate(context, R.layout.character_list_item, null);
```



Specify the context of where this should be inflated.

Specify XML TEMPLATE, which you created

AttachToRoot

Creating and returning the View

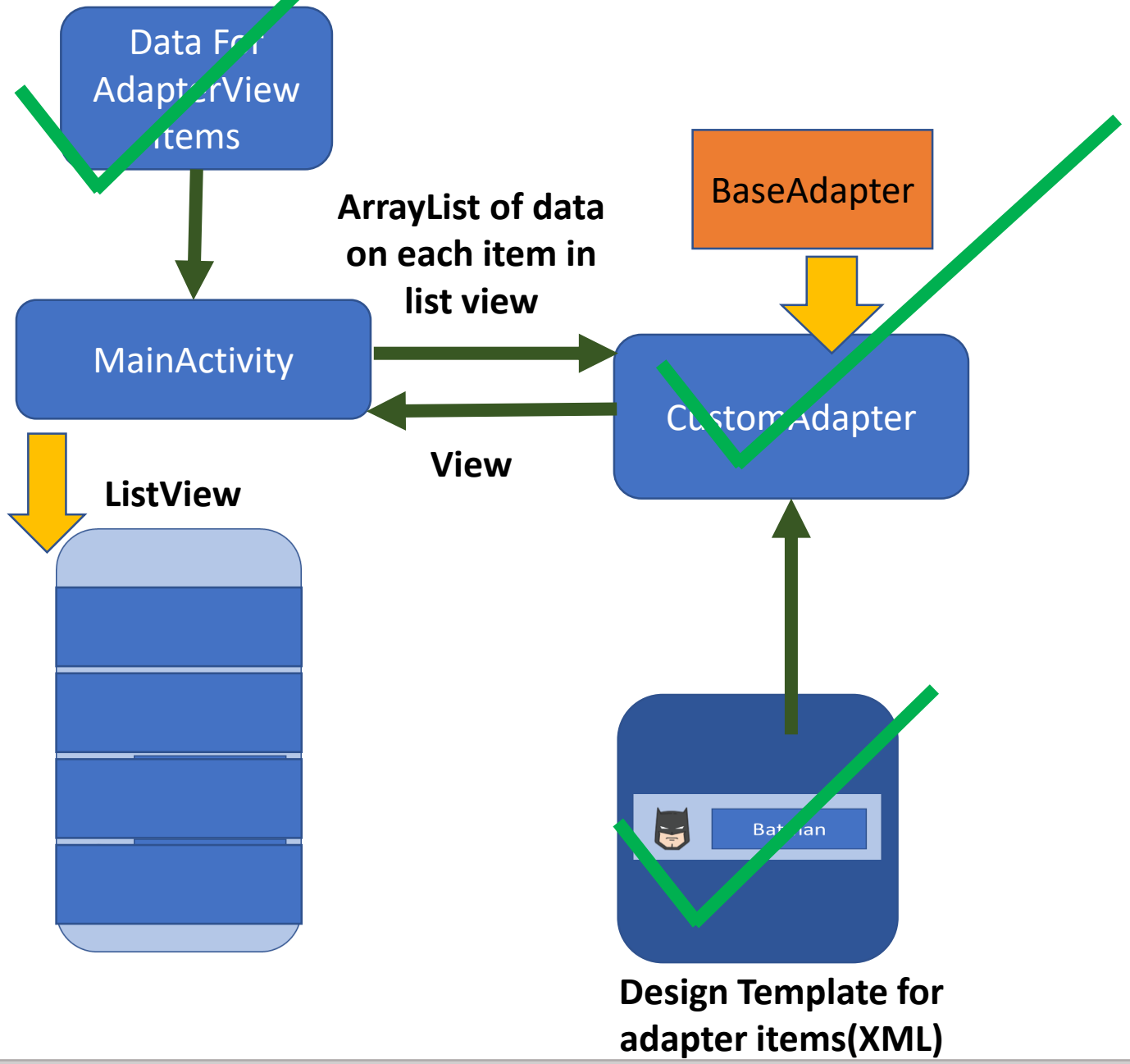
- Extract Children(ImageView and TextView) of view and assign values

- Return the view

Steps for creating Custom Adapter

- ~~Step 1 - Configuring the AdapterView Item~~
- ~~Step 2 - Creating CustomAdapter class~~
- ~~Step 3 - Implementing the BaseAdapter methods~~

- **Using your CustomAdapter in the MainActivity**
  - **Populate the data for AdapterView**
  - **Create a customAdapter**
  - **Set the CustomerAdapter to your ListView**
  - **Set clickListener to your listview**

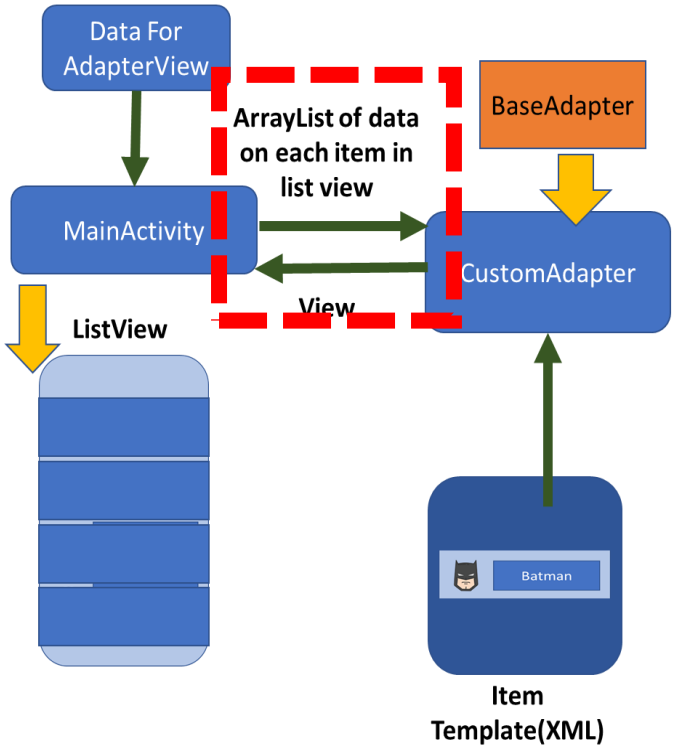

Data For AdapterView Items
ArrayList of data on each item in list view
BaseAdapter
MainActivity
Custom Adapter
View
ListView
Design Template for adapter items(XML)

Create Custom adapter and pass the data

1. Context – of the MainActivity

2. ArrayList - of type **FavoriteCharacterAdapterItem**

ImageResId	CharacterName
R.Drawable.batman	“Batman”
R.Drawable.joker	“Joker”
R.Drawable.spiderman	“Spiderman”
R.Drawable.thanos	“Thanos”

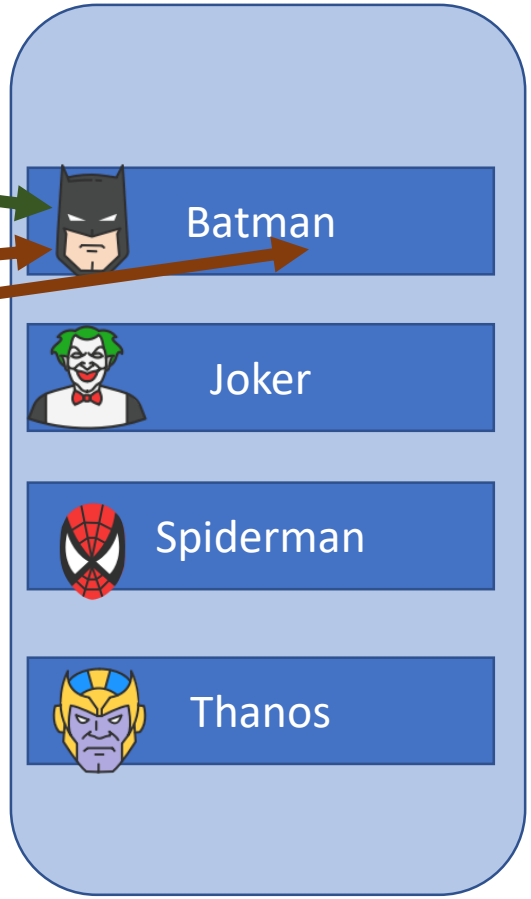


Finally!

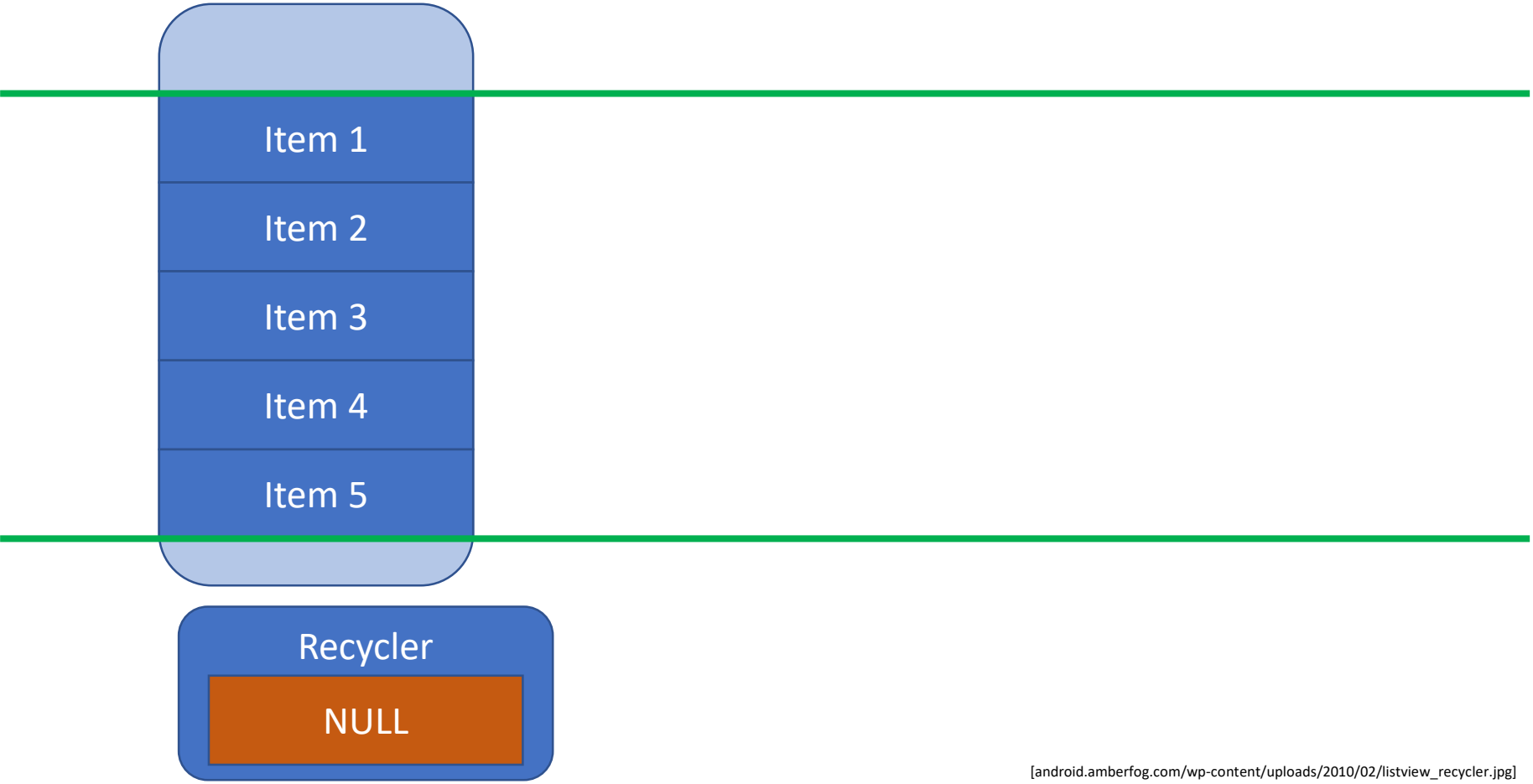
Is your current ListView Optimized?

Optimization?

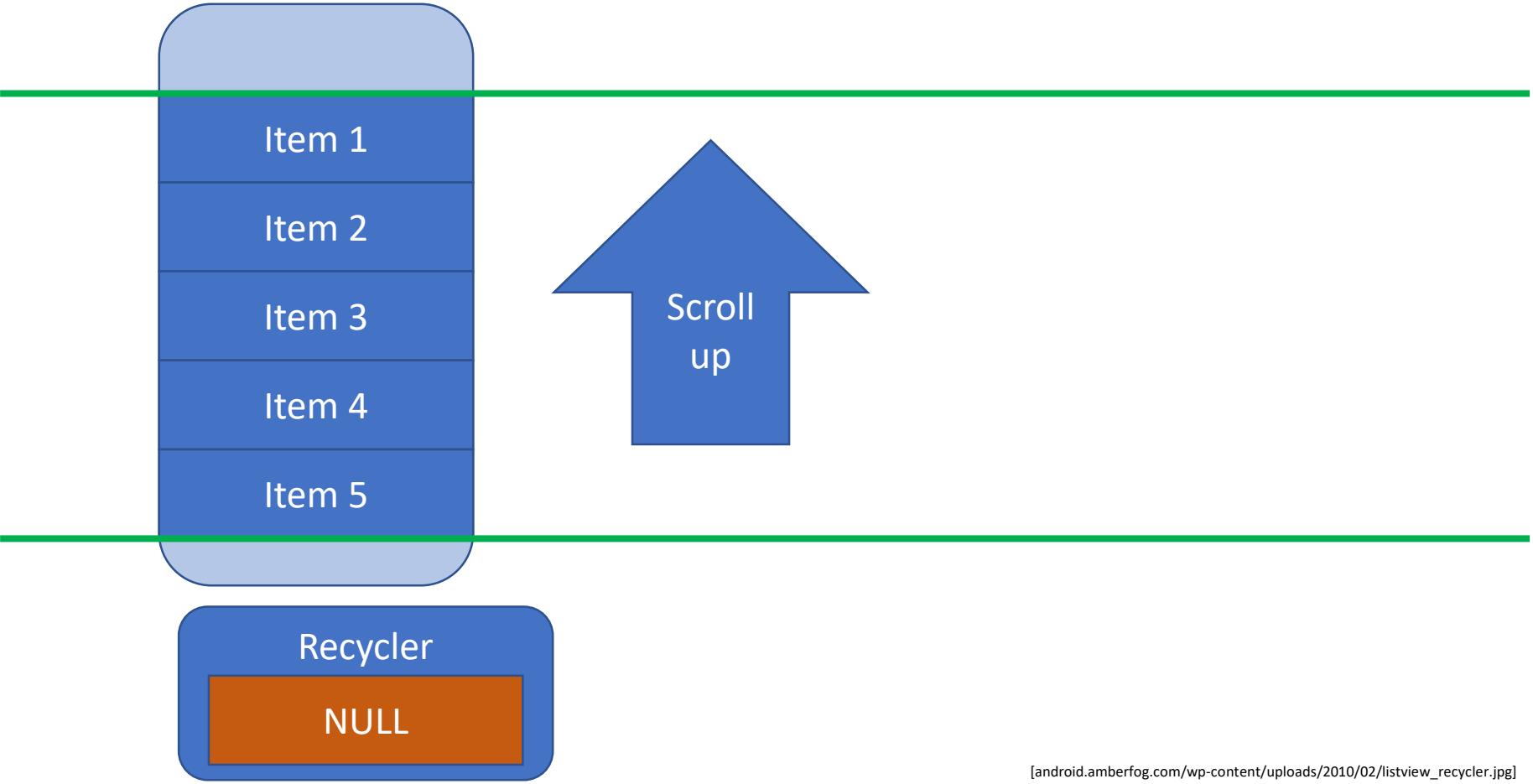
- Every time when a **VIEW** is created
  - Inflating the layout row
  - Finding the children of it
    - ImageView
    - TextView



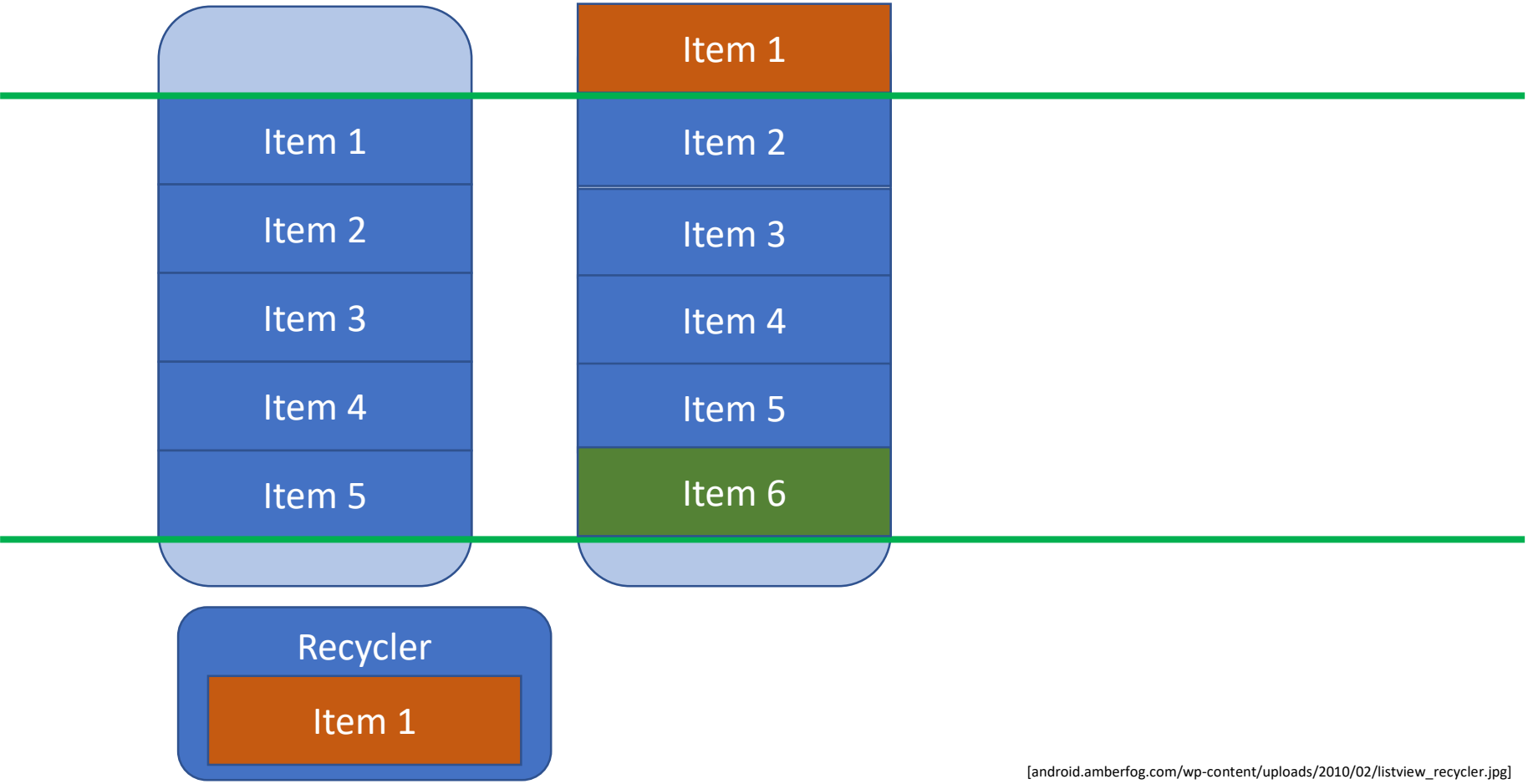
Optimizing ListViews



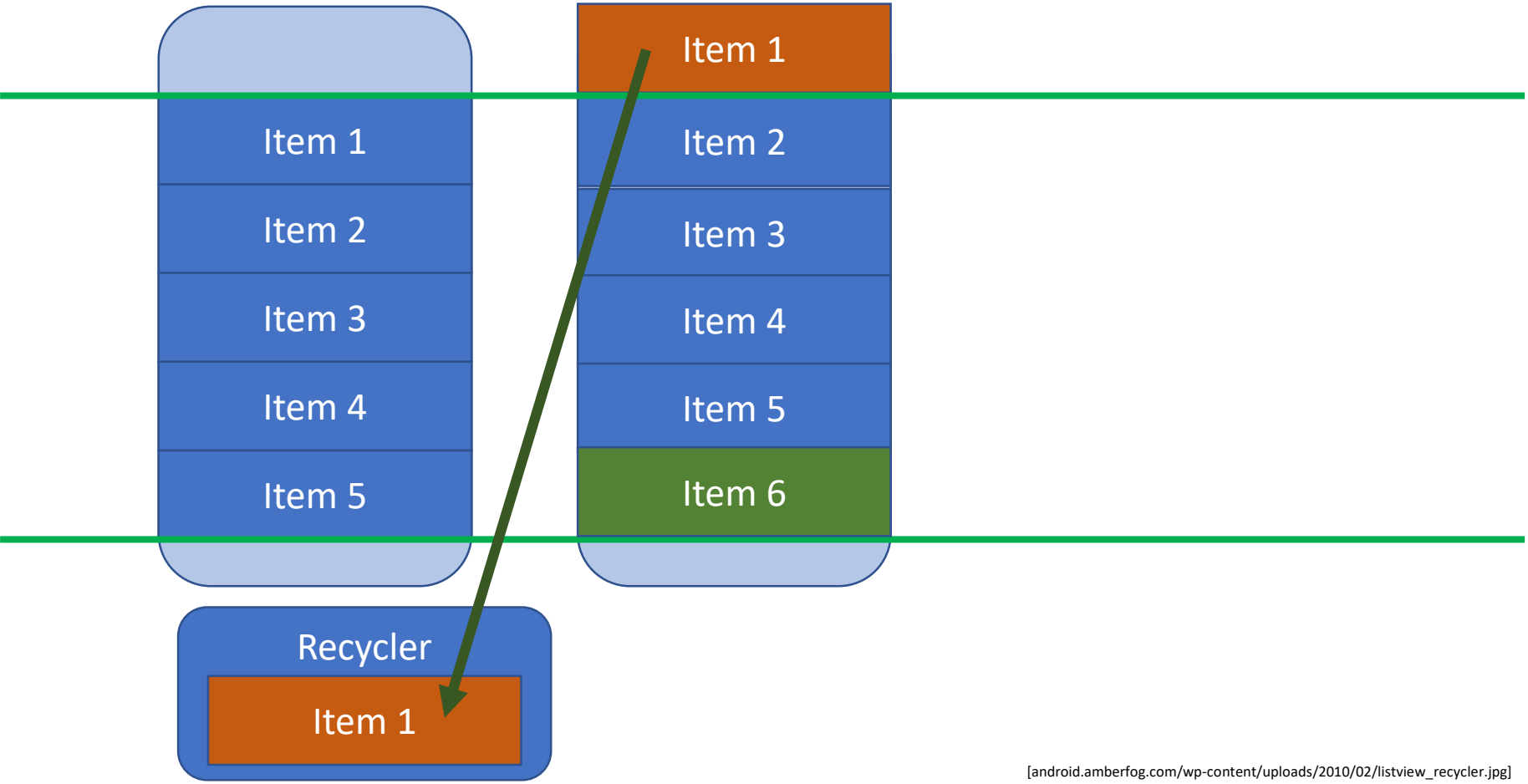
[android.amberfog.com/wp-content/uploads/2010/02/listview_recycler.jpg]



[android.amberfog.com/wp-content/uploads/2010/02/listview_recycler.jpg]

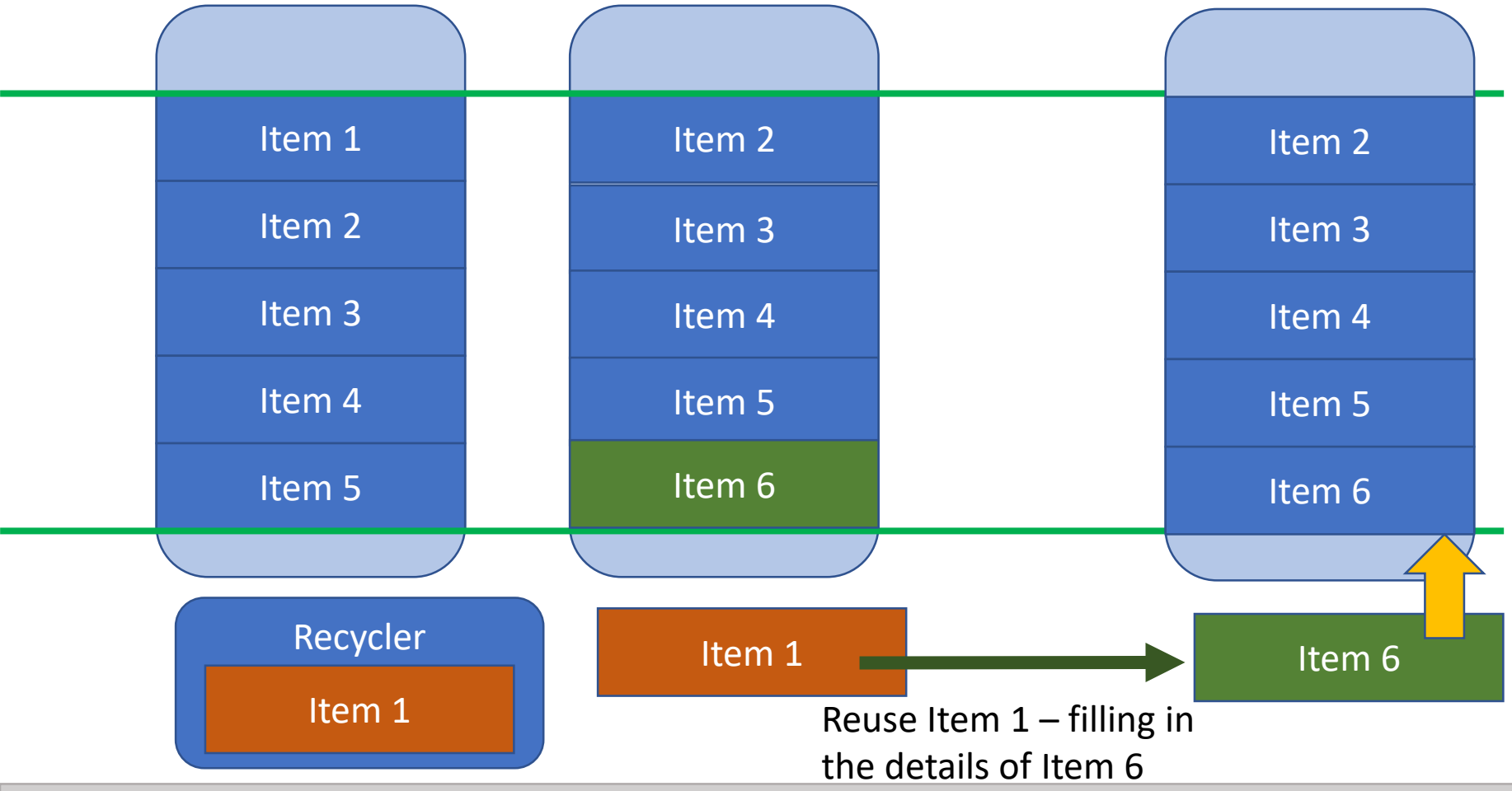


[android.amberfog.com/wp-content/uploads/2010/02/listview_recycler.jpg]

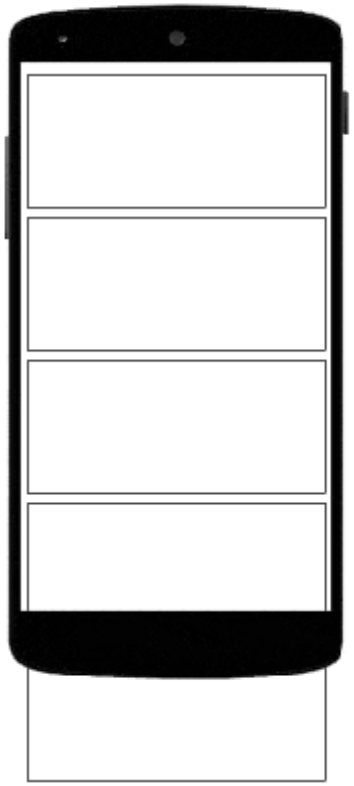


[android.amberfog.com/wp-content/uploads/2010/02/listview_recycler.jpg]

Recycling Views



Recycling views



[spreys.com/view-holder-design-pattern-for-android/]

Case 1 – The right way

- Reusing the view rather than creating a new one.
- View is NULL for the first time -> from second time you can reuse it by changing the contents of it

[youtube.com/watch?v=wDBM6wVEO70]

Optimization?

- Every time when a **VIEW** is created
  - **Inflating the layout row**
  - Finding the children of it
    - ImageView
    - TextView

Fixed!

???