

## MODULE

# RecyclerViews & Card View

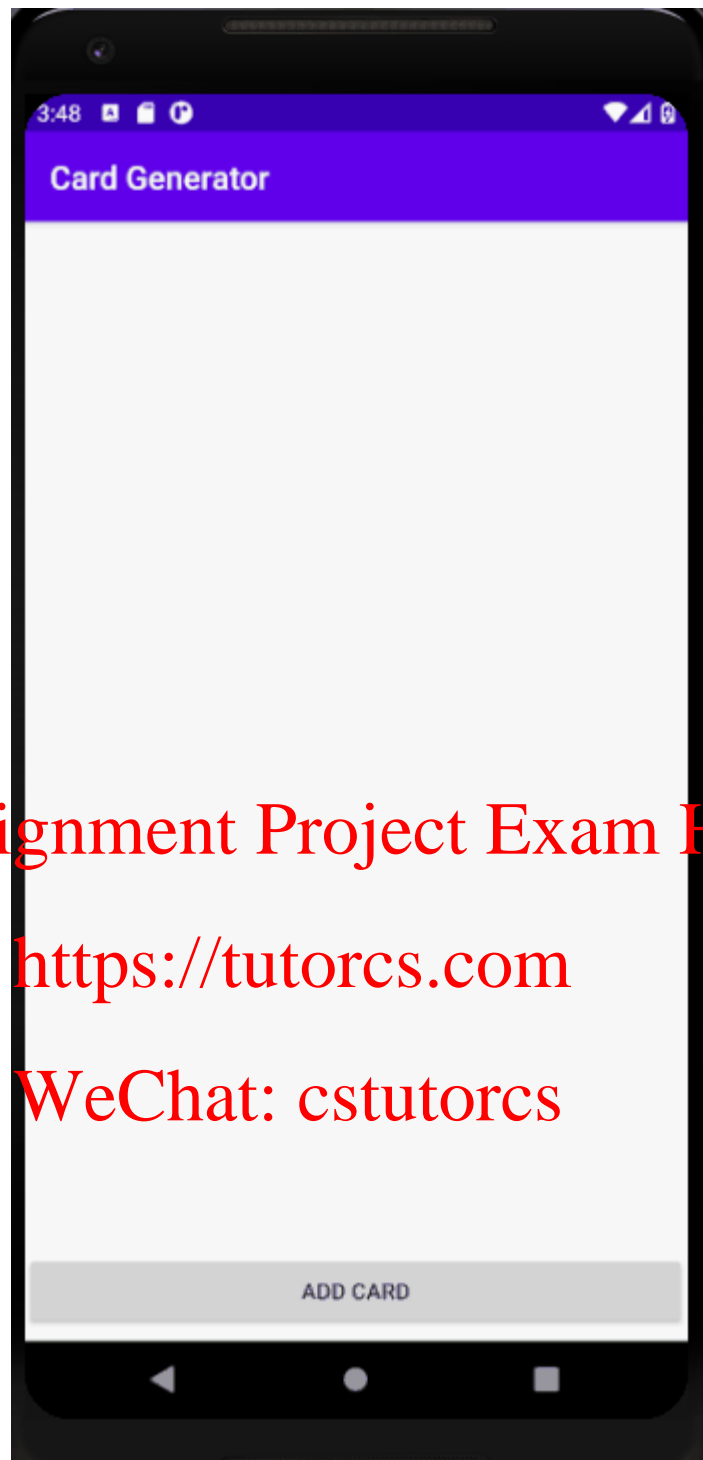
*Nawfal Ali**Updated 22 April 2020*

What is RecyclerView?

A RecyclerView is a flexible viewgroup for providing a limited window into a large data set. The purpose of the RecyclerView is to allow information to be presented to the user in the form of a scrollable list. The RecyclerView is significantly more efficient in the way it manages the views that make up a list, essentially reusing existing views that make up list items as they scroll off the screen instead of creating new ones (hence the name “recycler”).

## Random Card Generator Application

In the following sections, we are going to develop an application that uses RecyclerView to display Cards that are generated randomly. Each card is implemented as a CardView.



Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

## CardView

The CardView class is a user interface view that allows information to be presented in groups using a card metaphor. Cards are usually presented in lists using a RecyclerView instance and may be configured to appear with shadow effects and rounded corners.

The user interface layout to be presented with a CardView instance is defined within an XML layout resource file and loaded into the CardView at runtime. The CardView layout can contain a layout of any complexity using the standard layout managers such as RelativeLayout and LinearLayout.

## card\_layout.xml

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <androidx.cardview.widget.CardView
   xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:card_view="http://schemas.android.com/apk/res-auto"
4.     android:id="@+id/card_view"
5.     android:layout_width="match_parent"
6.     android:layout_height="wrap_content"
7.     android:layout_margin="5dp"
8.     card_view:cardBackgroundColor="@android:color/holo_blue_dark"
9.     card_view:cardCornerRadius="12dp">
10.
11.     <RelativeLayout
12.         android:layout_width="match_parent"
13.         android:layout_height="match_parent"
14.         android:padding="16dp">
15.
16.         <TextView
17.             android:id="@+id/suit_id"
18.             android:layout_width="wrap_content"
19.             android:layout_height="wrap_content"
20.             android:layout_alignParentTop="true"
21.             android:layout_marginStart="58dp"
22.             android:layout_marginTop="8dp"
23.             android:layout_toEndOf="@+id/card_id"
24.             android:textSize="30sp" />
25.
26.
```

```

27.         <TextView
28.             android:id="@+id/card_id"
29.             android:layout_width="wrap_content"
30.             android:layout_height="wrap_content"
31.             android:layout_alignParentStart="true"
32.             android:layout_alignParentTop="true"
33.             android:layout_marginStart="8dp"
34.             android:layout_marginTop="8dp"
35.             android:textSize="30sp" />
36.     </RelativeLayout>
37.
38. </androidx.cardview.widget.CardView>

```

As you can see, the cardview works as a top-level container that has a relative layout as a child element. Inside the relative layout we have two text views that will be used to display our data. You can replace the relative layout with any layout you do prefer.

<https://tutorcs.com>

## RecyclerView WeChat: cstutorcs

It is a container for rendering a larger data set of views that can be recycled and scrolled very efficiently. It uses a subclass of RecyclerView.Adapter for providing views that represent items in a data set.

activity\_main.xml

```

1.  <?xml version="1.0" encoding="utf-8"?>
2.  <androidx.constraintlayout.widget.ConstraintLayout
3.      xmlns:android="http://schemas.android.com/apk/res/android"
4.      xmlns:app="http://schemas.android.com/apk/res-auto"
5.      xmlns:tools="http://schemas.android.com/tools"
6.      android:layout_width="match_parent"
7.      android:layout_height="match_parent"
8.      tools:context=".MainActivity">

```

```
<androidx.constraintlayout.widget.Guideline
```

```
    android:id="@+id/guideline"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.91" />
```

```
<Button
```

```
    android:id="@+id/add_item"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Add card"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@id/guideline" />
```

```
<androidx.recyclerview.widget.RecyclerView
```

```
    android:id="@+id/my_recycler_view"
    android:scrollbars="vertical"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toTopOf="@+id/guideline"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Assignment Project Exam Help  
<https://tutores.com>  
WeChat: cstutores

The code provided above represents the layout file of the main activity. It uses constraint layout as the main container (@line2). The horizontal guideline is used to separate the button from the recycler view.

### Item.java

```
1. package com.fit2081.recyclercard;
2.
3. public class Item {
4.
5.     private String suit;
6.     private String card;
7.
8.     public Item(String suit, String card) {
9.         this.suit = suit;
10.        this.card = card;
11.    }
12.
13.    public String getSuit() {
14.        return suit;
15.    }
16.
17.    public String getCard() {
18.        return card;
19.    }
20. }
```

Each card consists of a suit and a card number. Item.java is a class that works as a record for one card. Later in the code, we will create an array list of cards using this class. The constructor is used to initialize the two instance variables while the two getters are used to retrieve their values.

## RecyclerView.Adapter

The adapter is created as a subclass of the RecyclerView.Adapter class and must, at a minimum, implement the following methods, which will be called at various points by the RecyclerView object to which the adapter is assigned:

- **getItemCount()** – This method must return a count of the number of items that are to be displayed in the list.
- **onCreateViewHolder()** – This method creates and returns a ViewHolder object initialized with the view that is to be used to display the data. This view is typically created by inflating the XML layout file.
- **onBindViewHolder()** – This method is passed the ViewHolder object created by the onCreateViewHolder() method together with an integer value indicating the list item that is about to be displayed. Contained within the ViewHolder object is the layout assigned by the onCreateViewHolder() method. It is the responsibility of the onBindViewHolder() method to populate the views in the layout with the text and graphics corresponding to the specified item and to return the object to the RecyclerView where it will be presented to the user.

Assignment Project Exam Help

<https://tutorcs.com>

MyRecyclerViewAdapter.java

WeChat: estutorcs

```
1. package com.film2081.recyclerviewcardview;
2.
3. import android.util.Log;
4. import android.view.LayoutInflater;
5. import android.view.View;
6. import android.view.ViewGroup;
7. import android.widget.TextView;
8.
9. import androidx.annotation.NonNull;
10. import androidx.recyclerview.widget.RecyclerView;
11.
12. import java.util.ArrayList;
13.
14. public class MyRecyclerViewAdapter extends
    RecyclerView.Adapter<MyRecyclerViewAdapter.ViewHolder> {
```

```
15.
16.     ArrayList<Item> data = new ArrayList<Item>();
17.
18.     public void setData(ArrayList<Item> data) {
19.         this.data = data;
20.     }
21.
22.     @NonNull
23.     @Override
24.     public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
25.         View v =
LayoutInflater.from(parent.getContext()).inflate(R.layout.card_layout,
parent, false); //CardView inflated as RecyclerView list item
26.         ViewHolder viewHolder = new ViewHolder(v);
27.         Log.d("week6App", "onCreateViewHolder");
28.         return viewHolder;
29.     }
30.
31.     @Override
32.     public void onBindViewHolder(@NonNull ViewHolder holder, int
position) {
33.         holder.cardTv.setText(data.get(position).getCard());
34.         holder.suitTv.setText(data.get(position).getSuit());
35.         Log.d("week6App", "onBindViewHolder");
36.
37.     }
38.
39.     @Override
40.     public int getItemCount() {
41.         return data.size();
42.     }
43.
44.     public class ViewHolder extends RecyclerView.ViewHolder {
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



```

45.         public TextView suitTv;
46.         public TextView cardTv;
47.
48.         public ViewHolder(@NonNull View itemView) {
49.             super(itemView);
50.             suitTv = itemView.findViewById(R.id.suit_id);
51.             cardTv = itemView.findViewById(R.id.card_id);
52.         }
53.     }
54. }

```

The code provided is the implementation of the RecyclerViewAdapter. It implements the three methods mentioned earlier and the ViewHolder subclass (@line44). The data is passed to the adaptor through a method called **setData** (@line18) which is called from the MainActivity.java (@line44). The method **getItemCount** (@line40) returns the size of the array list which is the number of items to be displayed in the list. The method **onCreateViewHolder** inflates the card's layout that we have implemented earlier each time a new card is required (@line25). It passes v, which is a reference to the card layout to the constructor of ViewHolder local class. At line 28, the method returns the view holder object that will be an input to the next method **onBindViewHolder**. In the latter method, we receive two parameters, a view holder and a position. Our job is to retrieve the data at that position and place it in that view holder.

MainActivity.java

```

1. package com.fit2081.recyclercard;
2.
3. import androidx.appcompat.app.AppCompatActivity;
4. import androidx.recyclerview.widget.LinearLayoutManager;
5. import androidx.recyclerview.widget.RecyclerView;
6.
7. import android.os.Bundle;
8. import android.view.View;
9. import android.widget.Button;

```

```

10.
11. import java.util.ArrayList;
12. import java.util.Random;
13.
14. public class MainActivity extends AppCompatActivity {
15.
16.     ArrayList<Item> data = new ArrayList<>();
17.     String suits[] = {"Hearts", "Diamonds", "Clubs", "Spade"};
18.     String cards[] = {"2", "3", "4", "5", "6", "7", "8", "9", "10",
"Jack", "Queen", "King", "Ace"};
19.
20.     Button btn;
21.     RecyclerView recyclerView;
22.     RecyclerView.LayoutManager layoutManager;
23.     MyRecyclerViewAdapter adapter;
24.
25.     @Override
26.     protected void onCreate(Bundle savedInstanceState) {
27.         super.onCreate(savedInstanceState);
28.         setContentView(R.layout.activity_main);
29.         btn = findViewById(R.id.add_item);
30.         btn.setOnClickListener(new View.OnClickListener() {
31.             @Override
32.             public void onClick(View v) {
33.                 addItem();
34.             }
35.         });
36.
37.         recyclerView = findViewById(R.id.my_recycler_view);
38.
39.         layoutManager = new LinearLayoutManager(this); //A
RecyclerView.LayoutManager implementation which provides similar
functionality to ListView.

```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

```
40.         recyclerView.setLayoutManager(layoutManager);    // Also
StaggeredGridLayoutManager and GridLayoutManager or a custom Layout
manager
41.
42.
43.         adapter = new MyRecyclerAdapter();
44.         adapter.setData(data);
45.         recyclerView.setAdapter(adapter);
46.
47.     }
48.
49.     public void addItem() {
50.         Random random = new Random();
51.         int randCard = random.nextInt(cards.length);
52.         int randSuit = random.nextInt(suits.length);
53.         Item item = new Item(suits[randSuit], cards[randCard]);
54.         data.addItem(item);
55.         adapter.notifyDataSetChanged();
56.
57.     }
58.
59. }
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

In the code above, we have created three entities, a recycler view (@line37), dataset (@line16), and an adapter (@line43). The adapter works as a bridge between the recyclerview and the dataset (@line43,44).

Each time the data gets changed, you should notify the adapter by calling `adapter.notifyDataSetChanged()` in order to update the recycler view (i.e. the UI)

## References:

- Smyth, Neil. Android Studio 3.5 Development Essentials – Java Edition: Developing Android 10 (Q) Apps Using Android Studio 3.5, Java and Android Jetpack . Payload Media, Inc.. Kindle Edition.
- <https://developer.android.com/>

---

Copyright © Monash University, unless otherwise stated. All Rights Reserved, except for individual components (or items) marked with their own licence restrictions



Copyright © 2021 Monash University, unless otherwise stated  
Disclaimer and Copyright

# Assignment Project Exam Help

Privacy

Service Status

## <https://tutorcs.com>

## WeChat: cstutorcs