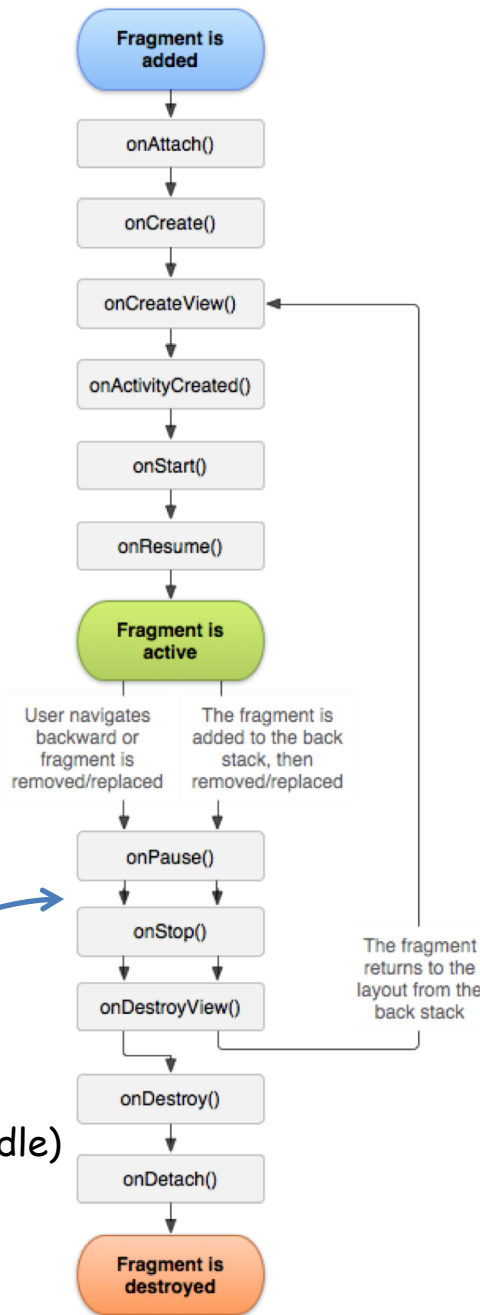# Android Fragments

CS 240 – Advanced Programming Concepts

# Fragments

- Reusable pieces of UI
- Layouts are already reusable, but fragments allow you to reuse the entire UI (including event handlers, etc.)
- Must be hosted by (displayed in) Activities
- Placed inside a **FrameLayout**
- Use the **FragmentManager** class to place a fragment in a **FrameLayout** of an Activity

# The Fragment Lifecycle

- Fragments have a lifecycle similar to activities
- Restore saved instance state in `onCreate(Bundle)` **or** `onCreateView(Bundle)`
- Save instance sate in `onSaveInstanceState(Bundle)` (same as Activity)
- Inflate the view in `onCreateView(…)`

**Fragment is added**

onAttach()

onCreate()

onCreateView()

onActivityCreated()

onStart()

onResume()

**Fragment is active**

User navigates backward or fragment is removed/replaced

The fragment is added to the back stack, then removed/replaced

onPause()

onStop()

onDestroyView()

The fragment returns to the layout from the back stack

onDestroy()

onSaveInstanceState(Bundle)

onDetach()

**Fragment is destroyed**

| Activity State | Fragment Callbacks |
|---|---|
| Created | onAttach() |
| | onCreate() |
| | onCreateView() |
| | onActivityCreated() |
| Started | onStart() |
| Resumed | onResume() |
| Paused | onPause() |
| Stopped | onStop() |
| Destroyed | onDestroyView() |
| | onDestroy() |
| | onDetach() |

Additional Info:
https://developer.android.com/guide/components/fragments

4

# Create a Fragment in an Activity

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

    ...

    FragmentManager fm = this.getSupportFragmentManager();
    shippingAddressFragment = (AddressFragment) fm.findFragmentById(R.id.shippingFrameLayout);
    if (shippingAddressFragment == null) {
        shippingAddressFragment = createAddressFragment(getString(R.string.shippingAddressTitle));
        fm.beginTransaction()
            .add(R.id.shippingFrameLayout, shippingAddressFragment)
            .commit();
    }
}
```

- Above code is in the hosting activity
- R.id.shippingFrameLayout is the ID of the FrameLayout we add the fragment
- Example: [FragmentExample](FragmentExample)/OrderInfoActivity.java

# Create a Fragment in an Activity

```
private AddressFragment createAddressFragment(String title) {
    AddressFragment fragment = new AddressFragment();

    Bundle args = new Bundle();
    args.putString(AddressFragment.ARG_TITLE, title);
    fragment.setArguments(args);

    return fragment;
}
```

# Inflate View in onCreateView(…)

- Fragments inflate their view in **onCreateView(…)**

```
@Override
public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                Bundle savedInstanceState) {
  View view = inflater.inflate(R.layout.fragment_address, container, false);

  TextView titleTextView = view.findViewById(R.id.titleTextView);
  titleTextView.setText(title);

  streetEditText = view.findViewById(R.id.streetEditText);
  cityEditText = view.findViewById(R.id.cityEditText);
  stateEditText = view.findViewById(R.id.stateEditText);
  zipCodeEditText = view.findViewById(R.id.zipCodeEditText);

  return view;
}
```

Always set to false. This would only be set to true if you were generating and connecting view objects in code (which is never recommended).

# Passing Data to a Fragment

- Fragments shouldn't depend on a specific activity, so they shouldn't access their hosting fragment's data directly

- Two main steps to pass data to a fragment:

  1. Pass an intent extra when starting the hosting activity

  2. The hosting activity passes the data to the fragment when it creates the fragment

# Passing Data to a Fragment Example ([CriminalIntent-Chapt10](CriminalIntent-Chapt10))

1. Pass an intent extra when starting the hosting activity
   - `CrimeListFragment.CrimeHolder.onClick()`
2. The hosting activity retrieves the data from it's intent
   - `CrimeActivity.createFragment()`
   - `SingleFragmentActivity.onCreate(Bundle)`
3. The hosting activity stashes the data in a Bundle, creates the fragment and attaches the bundle to the fragment
   - `CrimeActivity.createFragment()`
4. The fragment retrieves the data from it's arguments bundle
   - `CrimeFragment.onCreate(Bundle)`

# Applications to Family Map Client

- **MainActivity.onCreate(…)**
  - Add LoginFragment if user not logged in
  - Add MapFragment if user is logged in
- **MainActivity (login success)**
  - Replace LoginFragment with MapFragment
- **EventActivity.onCreate(…)**
  - Activity receives Event or EventId as intent extra
  - Activity creates MapFragment and passes Event or EventId as argument
  - Add MapFragment