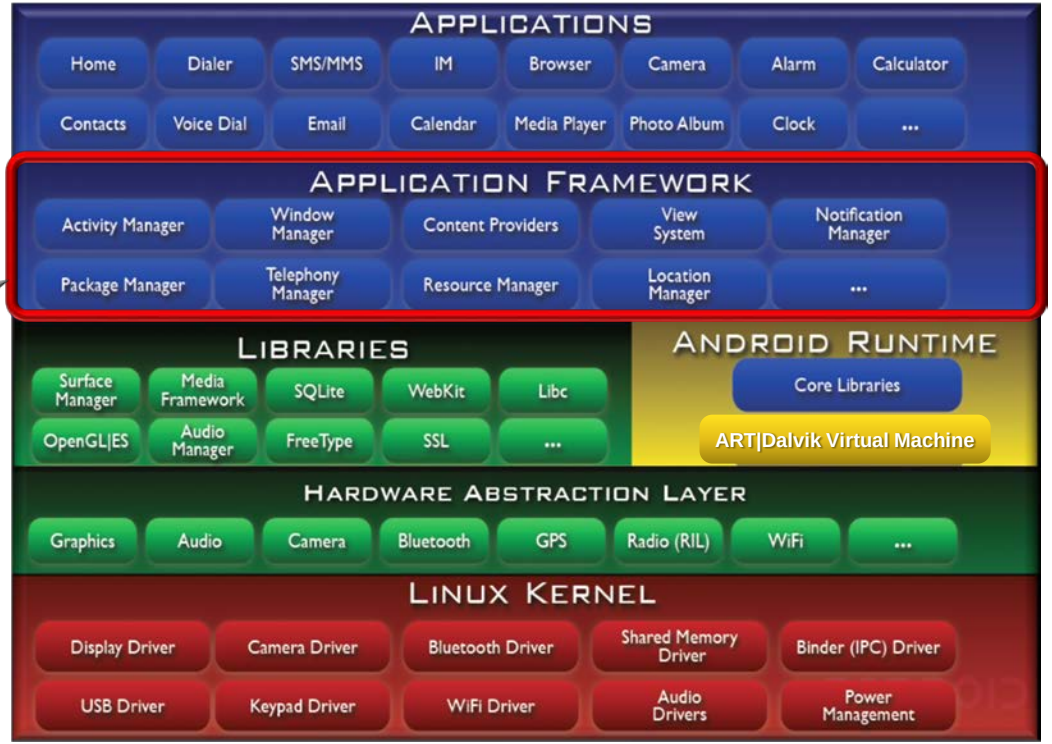
Android Common Services & Apps (Part 1): Overview of Object-Oriented Frameworks

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software
Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand what an object-oriented framework is in the context of Java & Android



Learning Objectives in this Part of the Lesson

- Understand what an object-oriented framework is in the context of Java & Android
- Recognize common examples of Java & Android frameworks

Learning Objectives in this Part of the Lesson

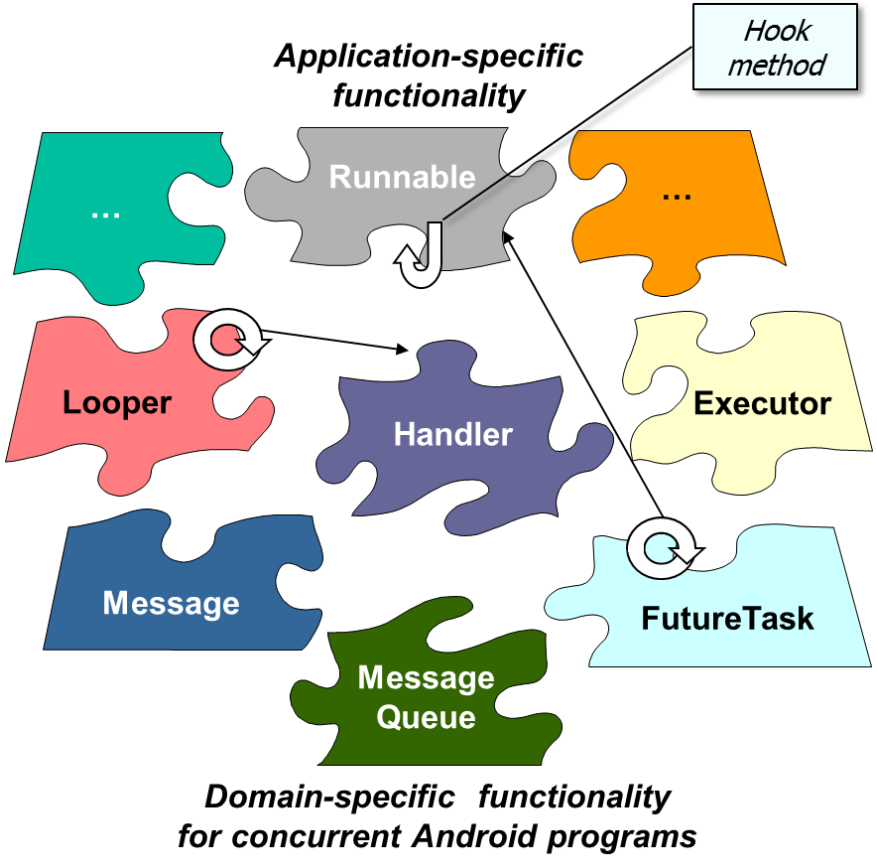
- Understand what an object-oriented framework is in the context of Java & Android
- Recognize common examples of Java & Android frameworks

This topic is important since *all* Android apps run within one or more frameworks

Overview of Object-Oriented Frameworks

Overview of Object-Oriented Frameworks

- A framework is an integrated set of components that provide a reusable architecture for a family of related applications



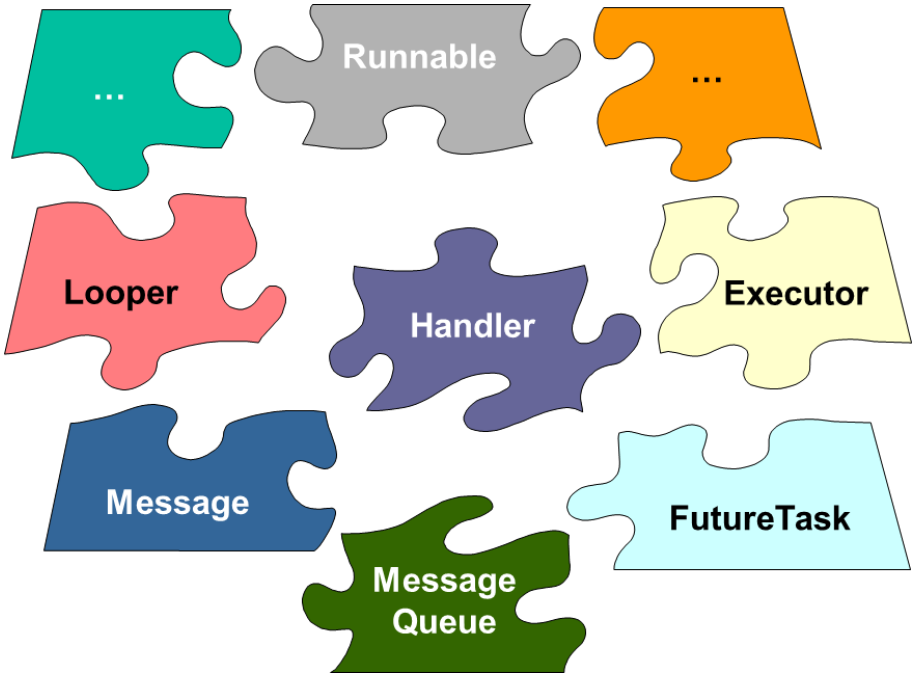
See www.dre.vanderbilt.edu/~schmidt/frameworks.html

Overview of Object-Oriented Frameworks

- Frameworks have 3 characteristics

Overview of Object-Oriented Frameworks

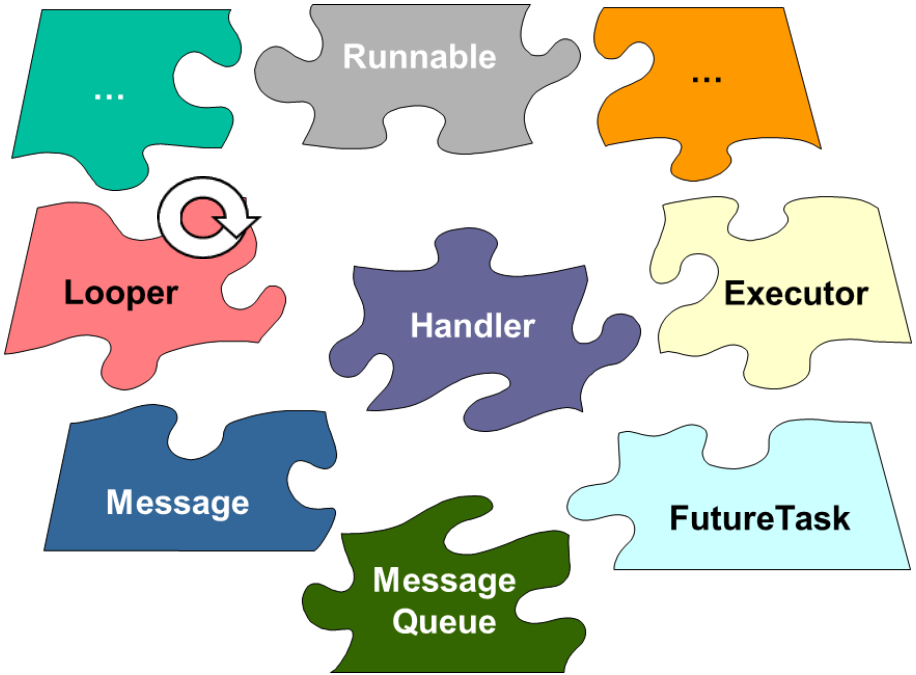
- Frameworks have 3 characteristics
  - Exhibit “inversion of control” (IoC) via callbacks



See en.wikipedia.org/wiki/Inversion_of_control

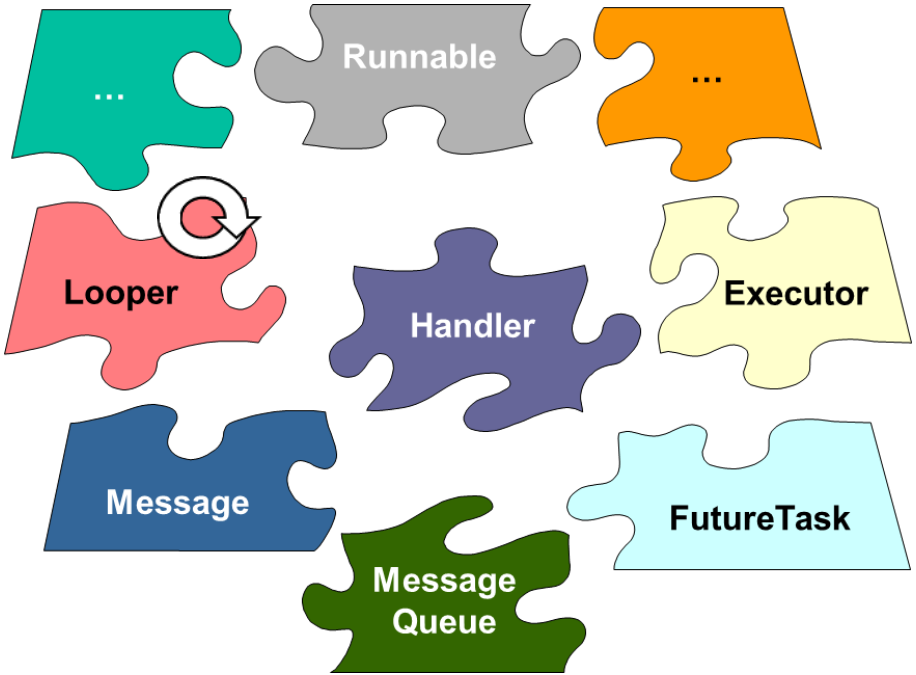
Overview of Object-Oriented Frameworks

- Frameworks have 3 characteristics
  - Exhibit “inversion of control” (IoC) via callbacks, i.e.
    - Framework controls the main execution thread



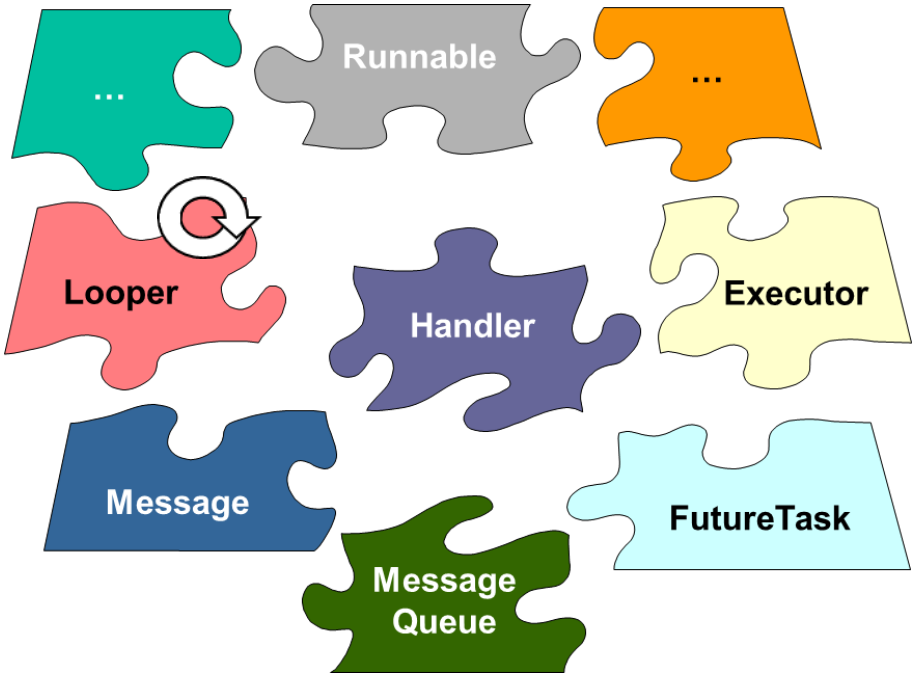
Overview of Object-Oriented Frameworks

- Frameworks have 3 characteristics
  - Exhibit “inversion of control” (IoC) via callbacks, i.e.
    - Framework controls the main execution thread
    - Decides how & when to run application code



Overview of Object-Oriented Frameworks

- Frameworks have 3 characteristics
  - Exhibit “inversion of control” (IoC) via callbacks, i.e.
    - Framework controls the main execution thread
    - Decides how & when to run application code
    - This inversion of control is often called the “Hollywood Principle”

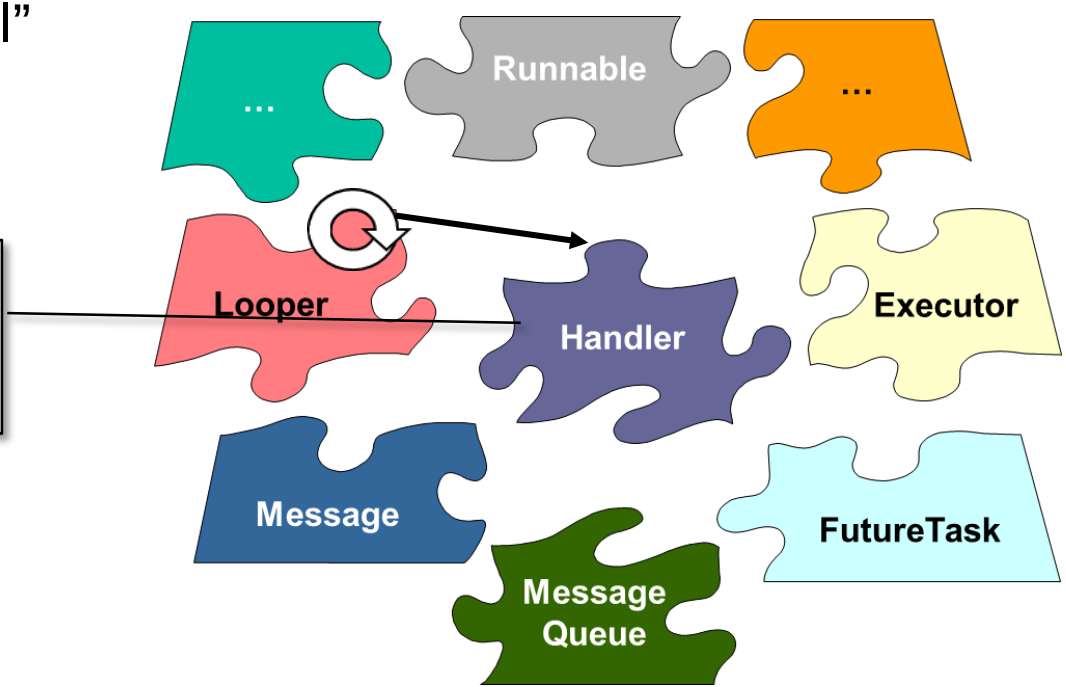


See www.dre.vanderbilt.edu/~schmidt/Coursera/articles/hollywood-principle.txt

Overview of Object-Oriented Frameworks

- Frameworks have 3 characteristics
  - Exhibit "inversion of control" (IoC) via callbacks, i.e.

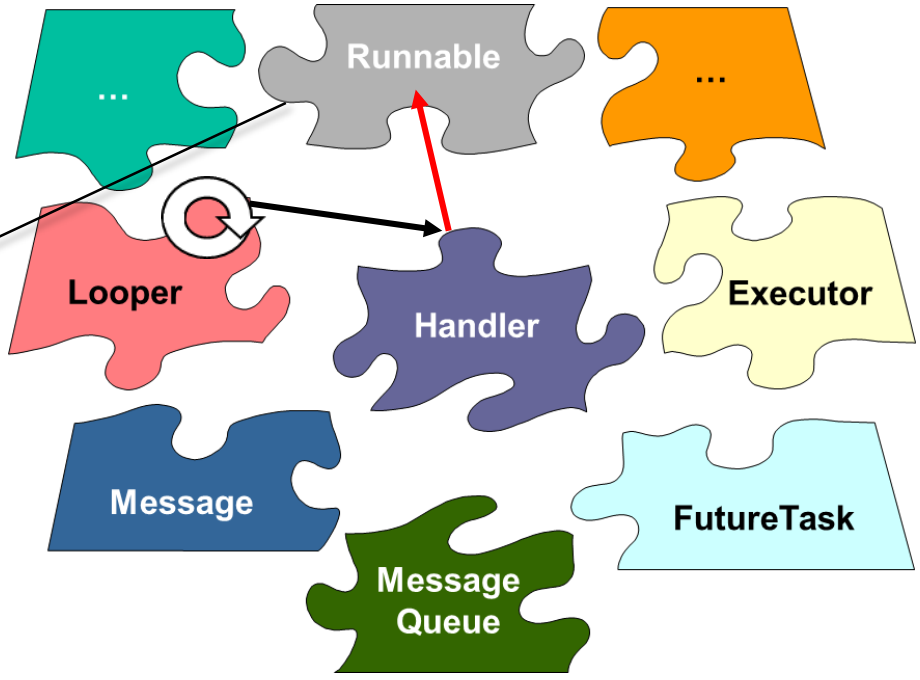
e.g., an Android looper dispatches a handler, which then dispatches a runnable



Overview of Object-Oriented Frameworks

- Frameworks have 3 characteristics
  - Exhibit "inversion of control" (IoC) via callbacks, i.e.

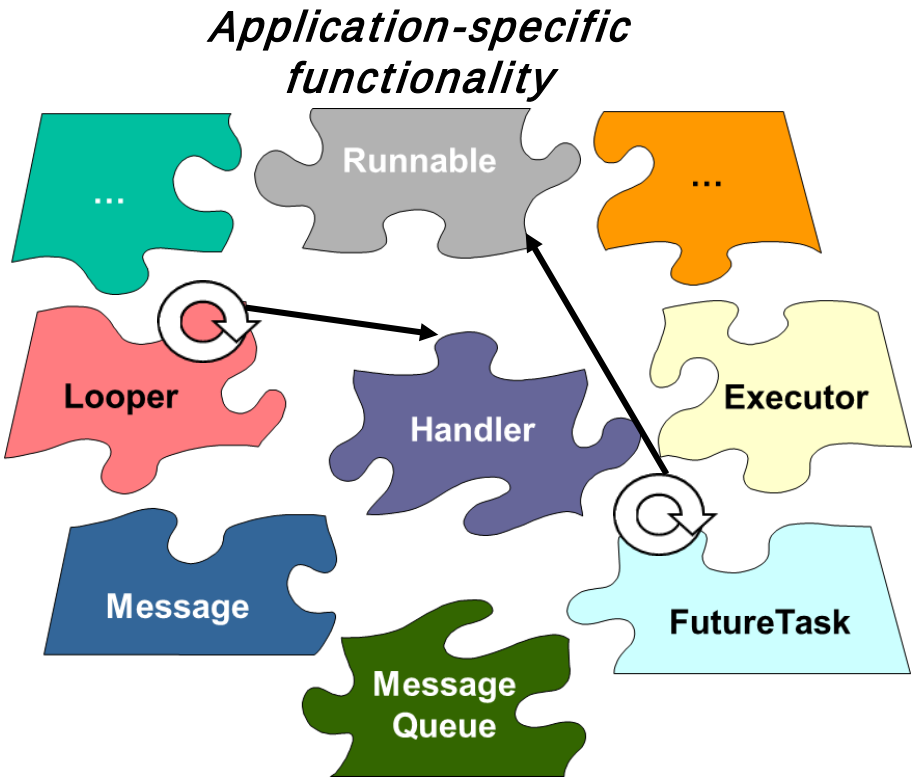
e.g., an Android looper dispatches a handler, which then dispatches a runnable



The runnable dispatched via IoC doesn't know/care how/why it was called back

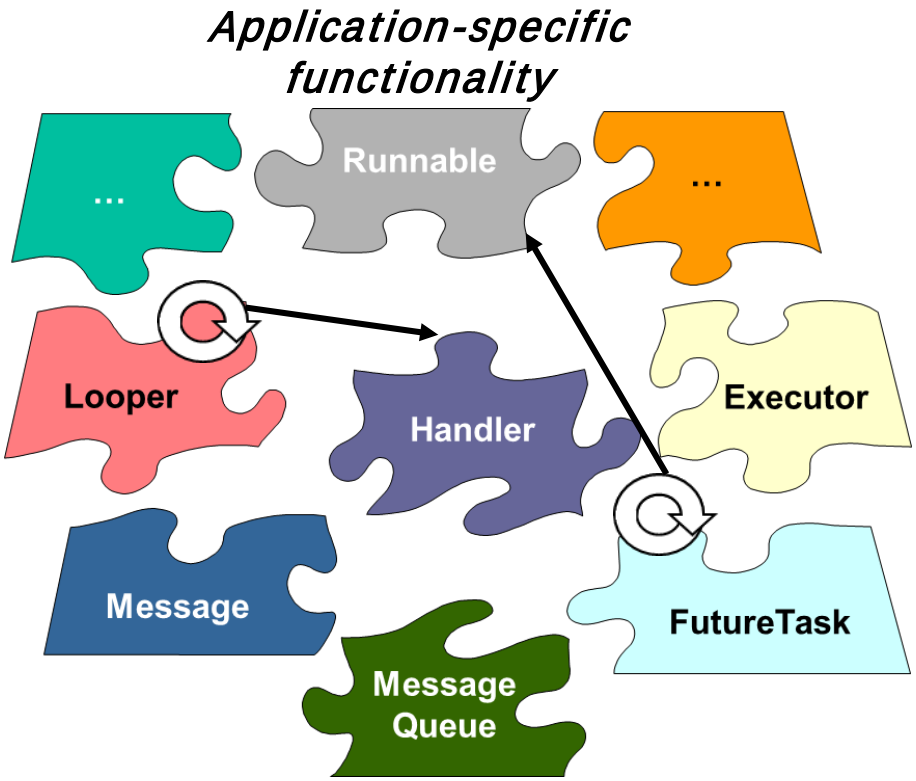
Overview of Object-Oriented Frameworks

- Frameworks have 3 characteristics
  - Exhibit "inversion of control" (IoC) via callbacks
  - Integrated domain-specific structure & functionality



Overview of Object-Oriented Frameworks

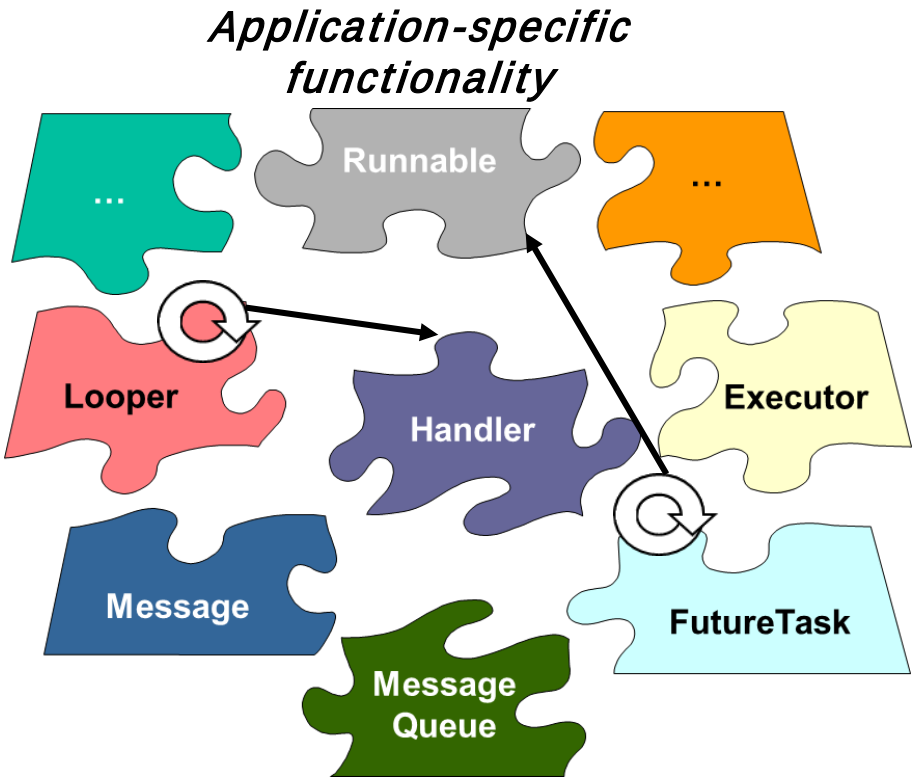
- Frameworks have 3 characteristics
  - Exhibit "inversion of control" (IoC) via callbacks
  - Integrated domain-specific structure & functionality
    - e.g., provide capabilities that can be reused in one or more domain(s)



Android's frameworks focus on domains associated with mobile apps & services

Overview of Object-Oriented Frameworks

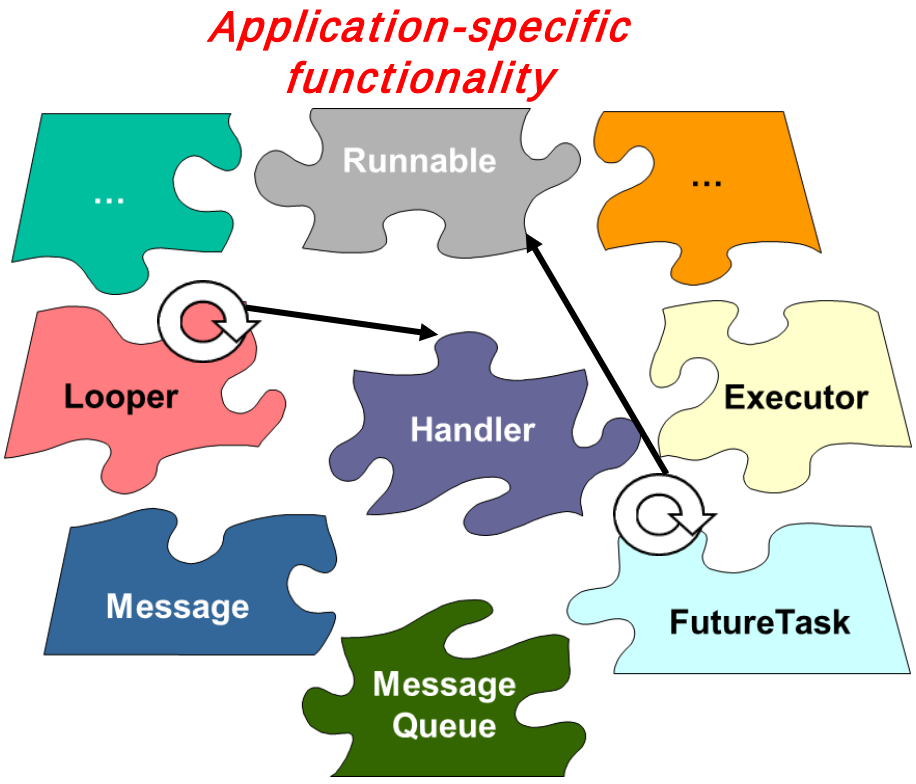
- Frameworks have 3 characteristics
  - Exhibit "inversion of control" (IoC) via callbacks
  - Integrated domain-specific structure & functionality
    - e.g., provide capabilities that can be reused in one or more domain(s)



Android's frameworks focus on domains associated with mobile apps & services

Overview of Object-Oriented Frameworks

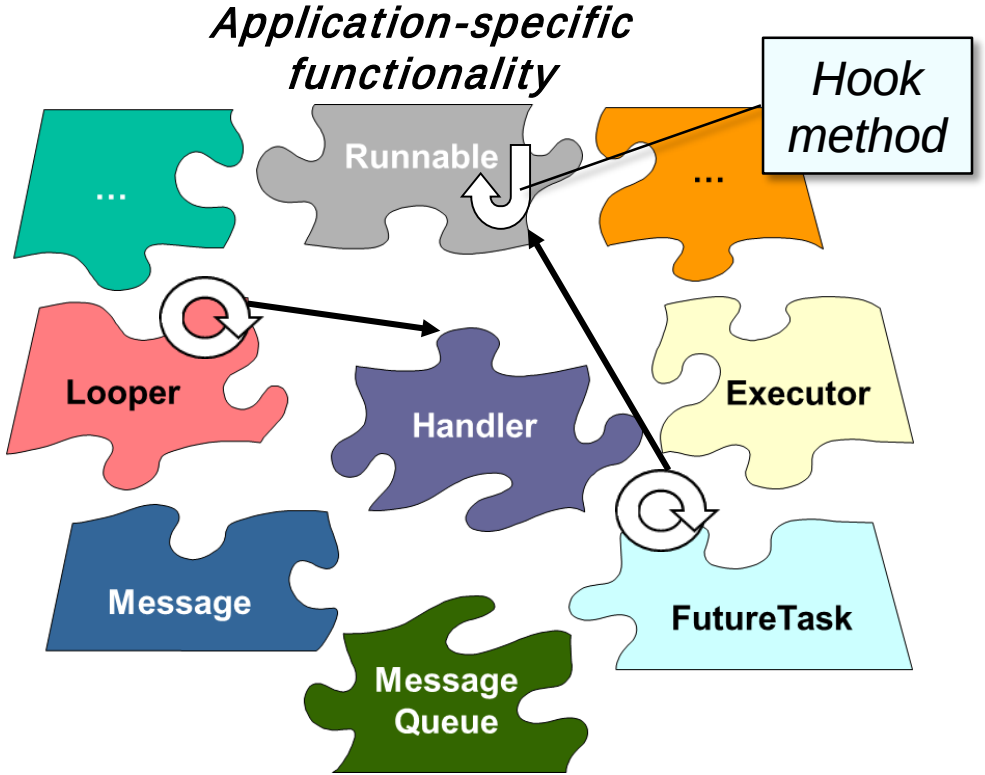
- Frameworks have 3 characteristics
  - Exhibit "inversion of control" (IoC) via callbacks
  - Integrated domain-specific structure & functionality
    - e.g., provide capabilities that can be reused in one or more domain(s)



Application-specific functionality can systematically reuse framework components

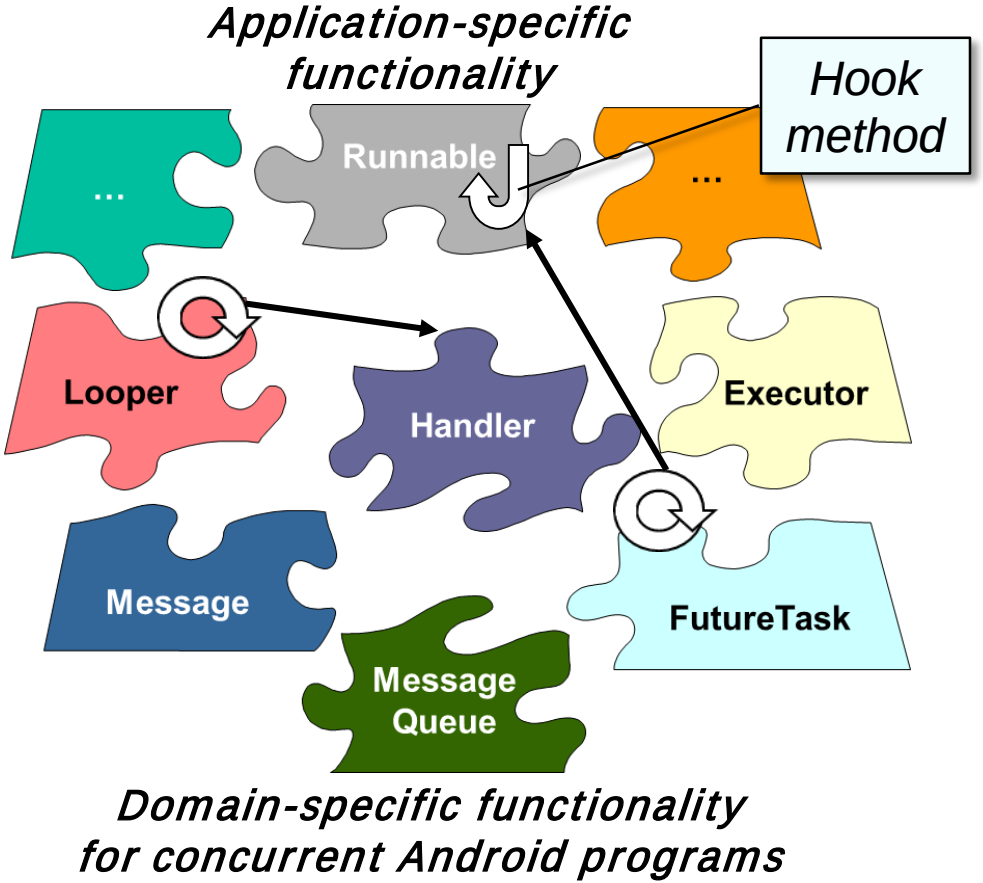
Overview of Object-Oriented Frameworks

- Frameworks have 3 characteristics
  - Exhibit "inversion of control" (IoC) via callbacks
  - Integrated domain-specific structure & functionality
  - Provide a "semi-complete application"



Overview of Object-Oriented Frameworks

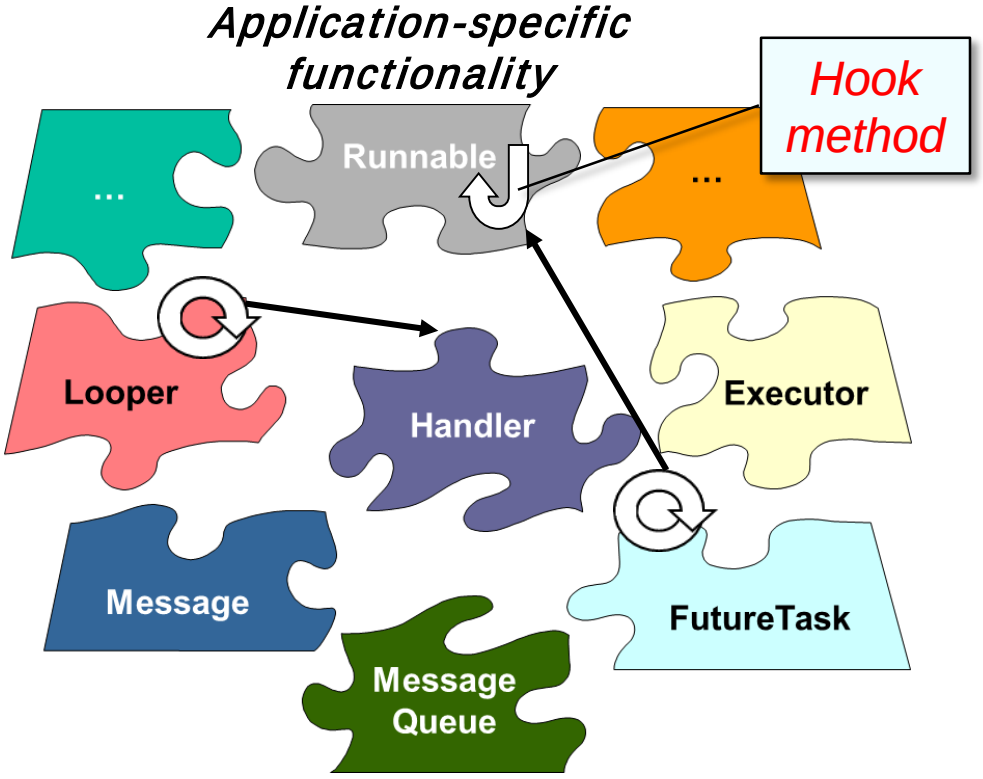
- Frameworks have 3 characteristics
  - Exhibit "inversion of control" (IoC) via callbacks
  - Integrated domain-specific structure & functionality
  - Provide a "semi-complete application"
    - *Hook methods* plug app logic into the framework



Hook methods implement patterns that customize framework components

Overview of Object-Oriented Frameworks

- Frameworks have 3 characteristics
  - Exhibit "inversion of control" (IoC) via callbacks
  - Integrated domain-specific structure & functionality
  - Provide a "semi-complete application"
    - *Hook methods* plug app logic into the framework
    - Mediate interactions among *common* abstract & *variant* concrete classes/interfaces



e.g., Java Runnable is an abstract interface providing basis for concrete variants

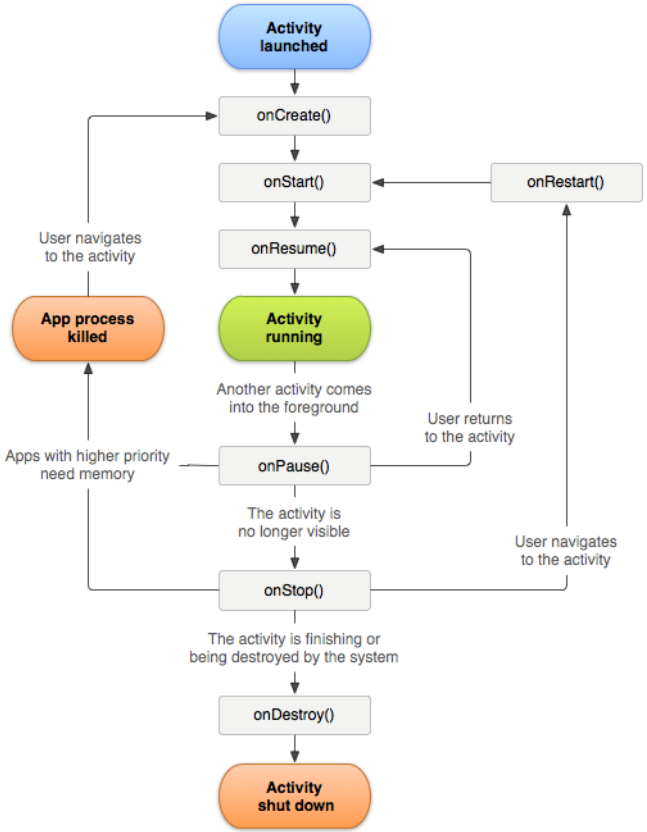
Overview of Frameworks in Android & Java

Overview of Frameworks in Android & Java

- Android & Java provide many frameworks

Overview of Frameworks in Android & Java

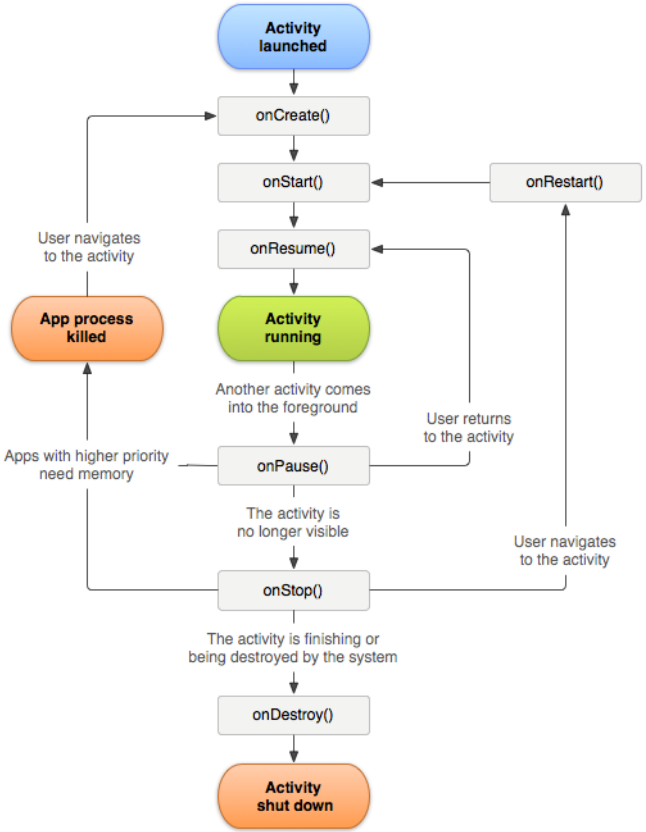
- Android & Java provide many frameworks
  - **Android**
    - Android activity framework controls the main thread



See developer.android.com/training/multiple-threads/communicate-ui.html

Overview of Frameworks in Android & Java

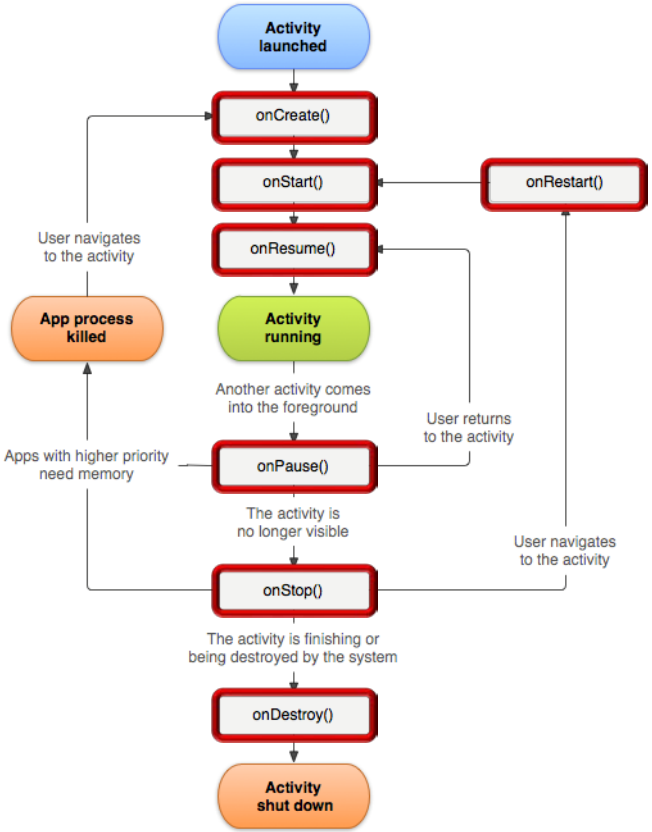
- Android & Java provide many frameworks
  - **Android**
    - Android activity framework controls the main thread
    - An application's lifecycle methods are called back by Android's activity framework



See developer.android.com/training/basics/activity-lifecycle

Overview of Frameworks in Android & Java

- Android & Java provide many frameworks
  - **Android**
    - Android activity framework controls the main thread
    - An application's lifecycle methods are called back by Android's activity framework
      - e.g., onCreate(), onStart(), onStop(), onDestroy(), etc.



See developer.android.com/training/basics/activity-lifecycle

Overview of Frameworks in Android & Java

- Android & Java provide many frameworks
  - **Android**
    - Android activity framework controls the main thread
    - An application's lifecycle methods are called back by Android's activity framework
    - A listener for button clicks is called back by Android's GUI framework

```
final Button button =
  (Button) findViewById
             (R.id.loadButton);
button.setOnClickListener
  (new OnClickListener() {
     @Override
     public void onClick(View v) {
       ...
  }});
```

A GUI component sending an event to its registered listener

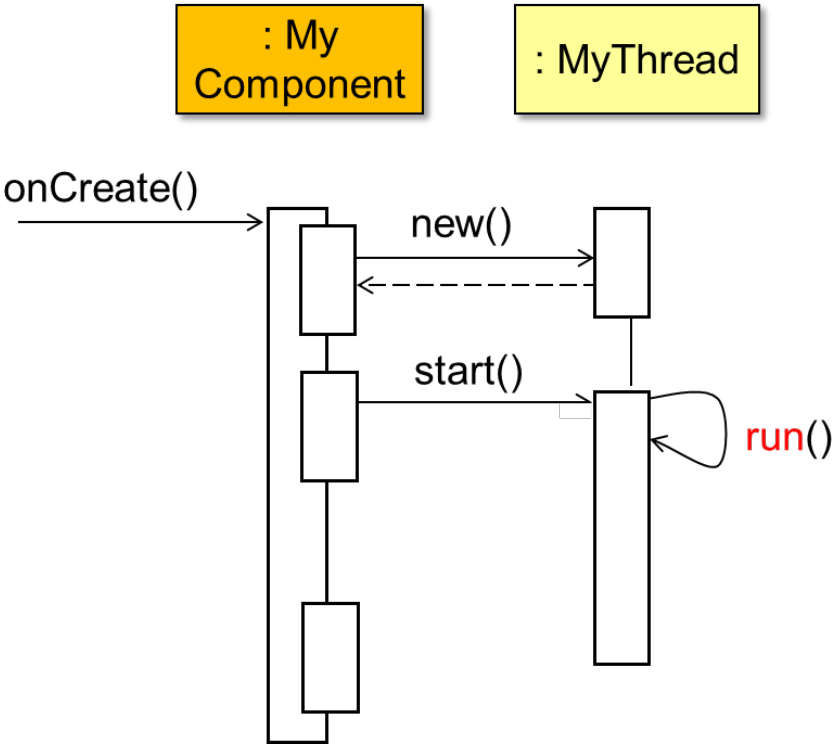
GUI Component (a button)

```
public void
onClick(View v){
  ...
}
```

ClickListener Object

Overview of Frameworks in Android & Java

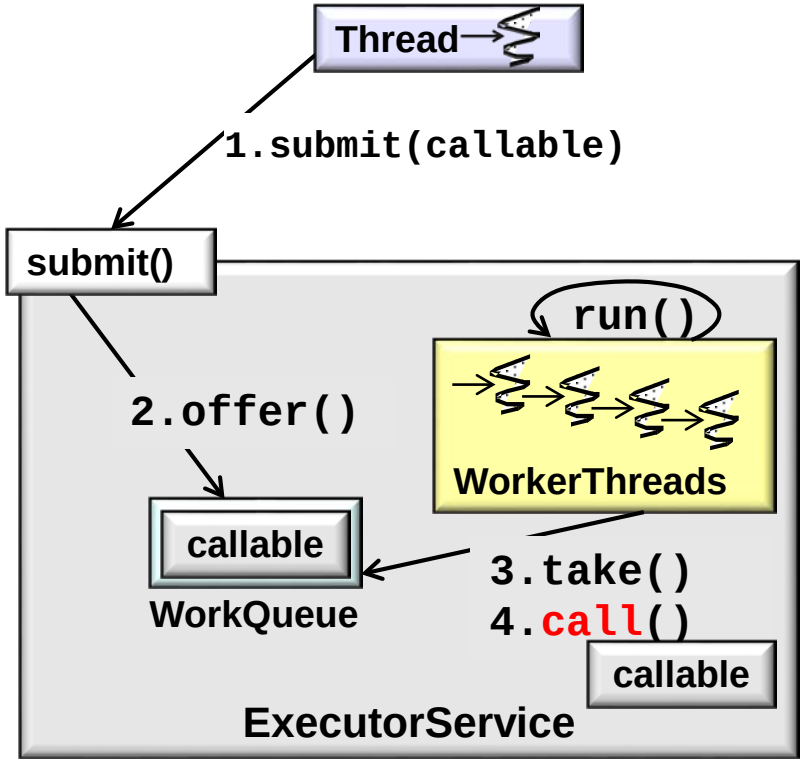
- Android & Java provide many frameworks
  - Android
  - **Java**
    - A thread calls back to the run() hook method of a Runnable



See docs.oracle.com/javase/tutorial/essential/concurrency/runthread.html

Overview of Frameworks in Android & Java

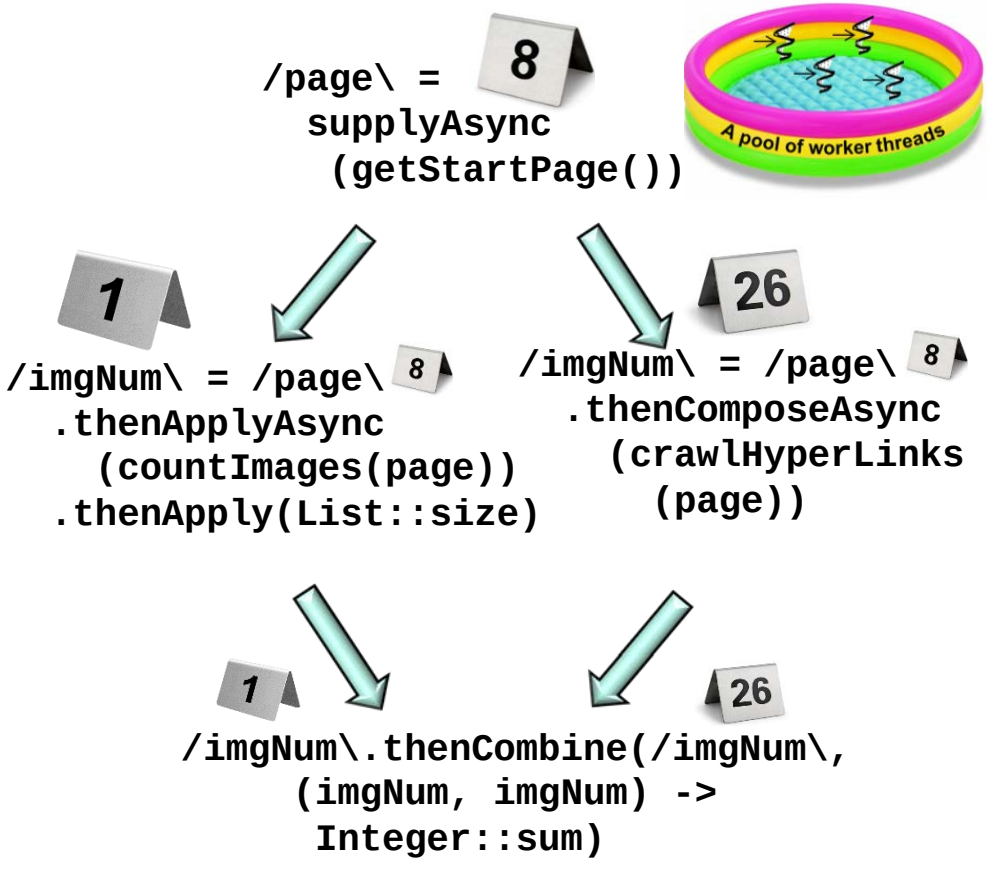
- Android & Java provide many frameworks
  - **Android**
  - **Java**
    - A thread calls back to the run() hook method of a Runnable
    - The ExecutorService calls back to the call() hook method of a Callable



See docs.oracle.com/javase/tutorial/essential/concurrency/executors.html

Overview of Frameworks in Android & Java

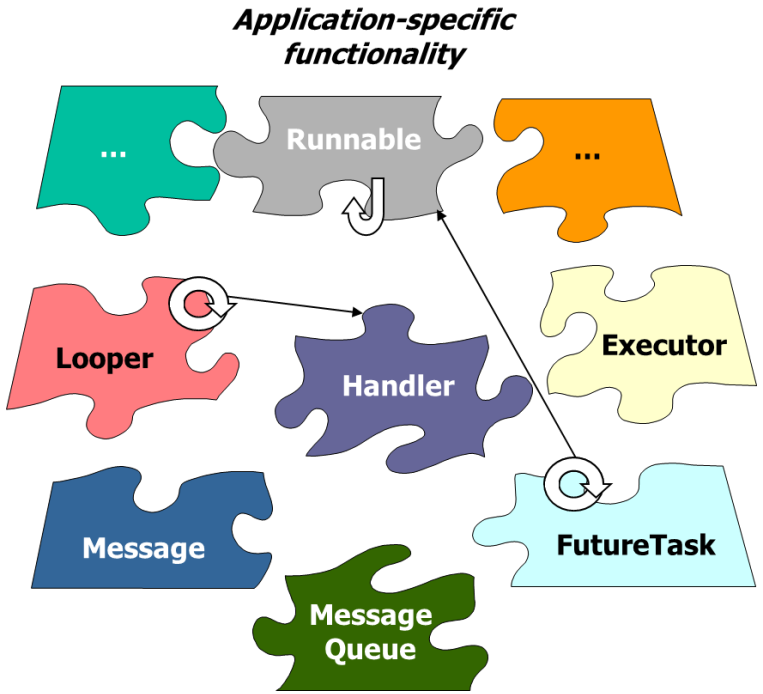
- Android & Java provide many frameworks
  - Android
  - **Java**
    - A thread calls back to the run() hook method of a Runnable
    - The ExecutorService calls back to the call() hook method of a Callable
    - Completable future & fork-join pool frameworks



See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html

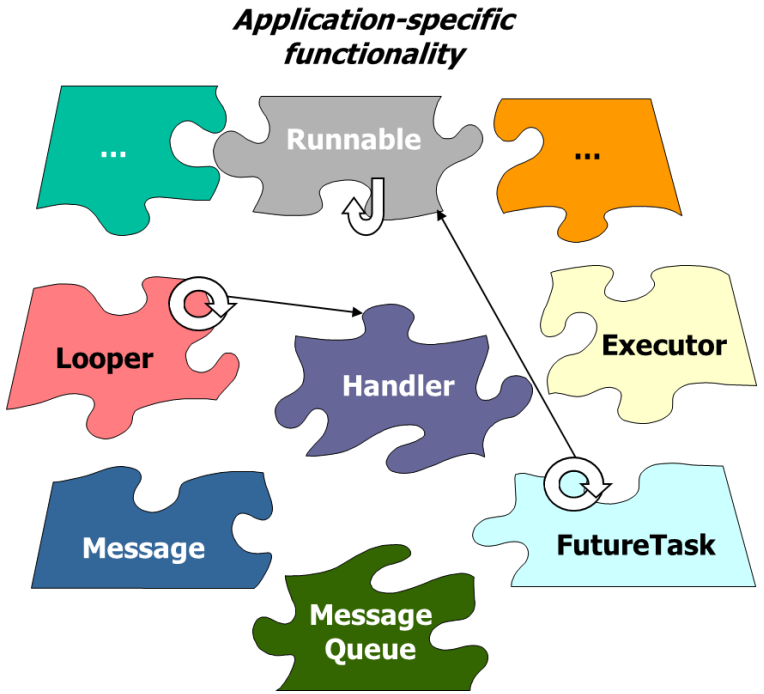
Overview of Frameworks in Android & Java

- Most code you write in this course will use one or more frameworks



Overview of Frameworks in Android & Java

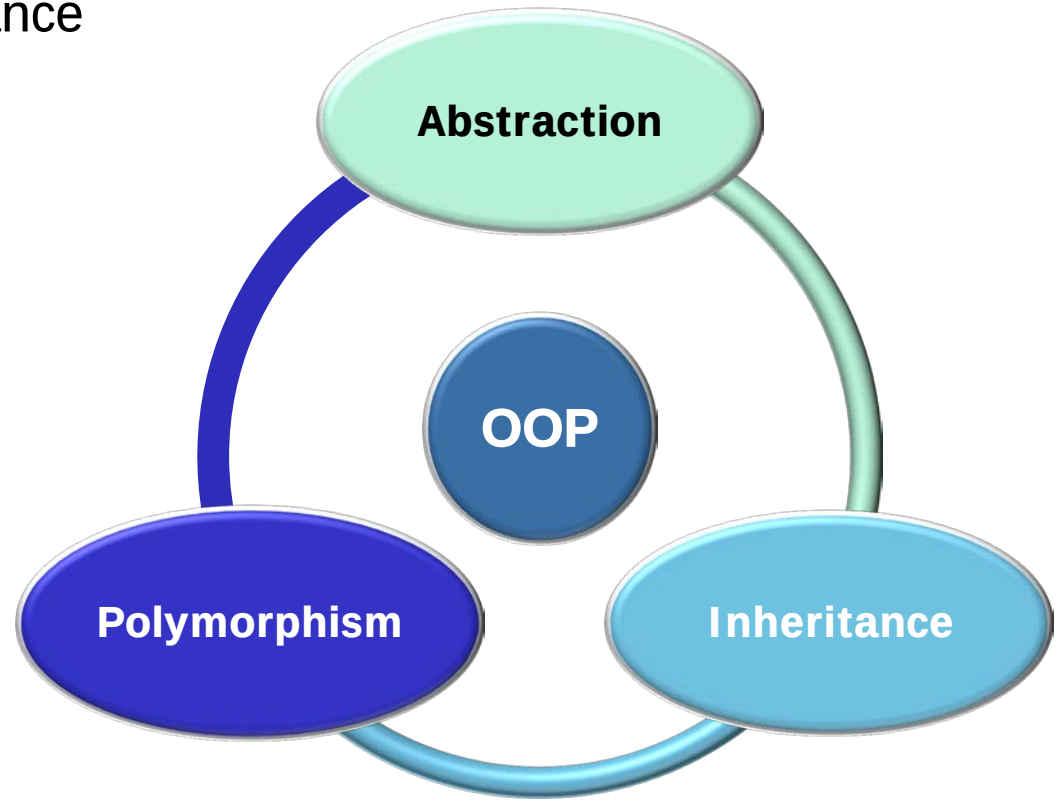
- Most code you write in this course will use one or more frameworks
  - Some of your solutions will implement your own framework



All Android apps run inside of one or more frameworks

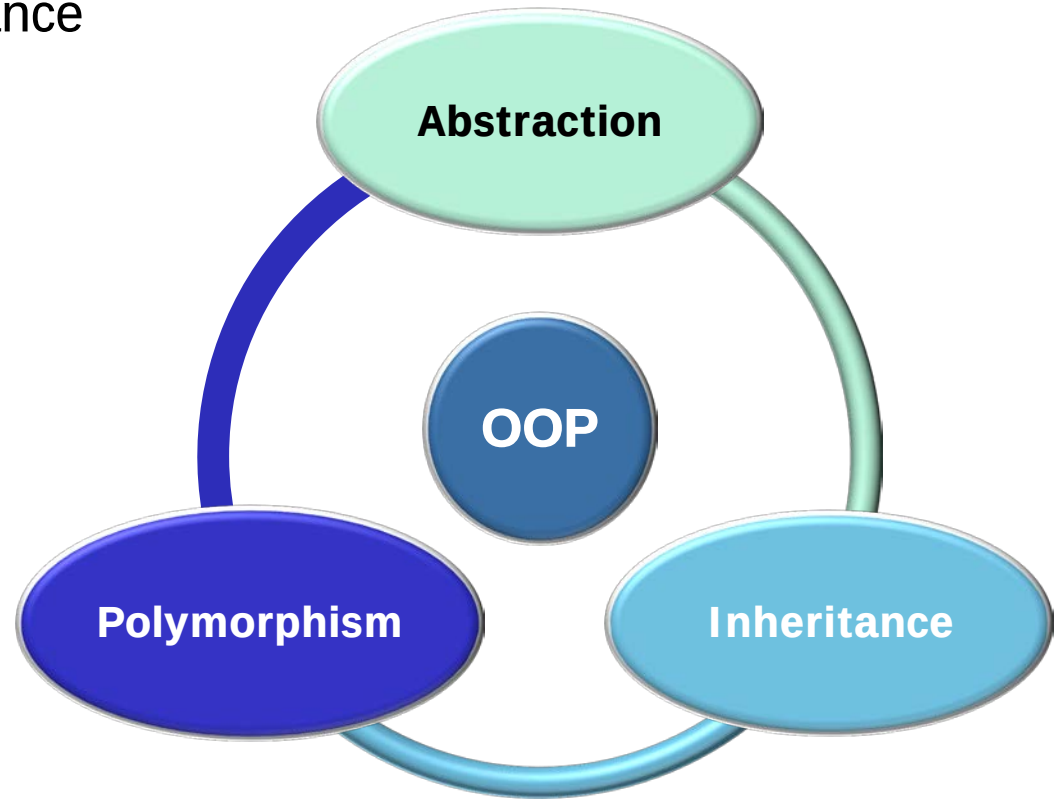
Overview of Frameworks in Android & Java

- Frameworks use Java's inheritance & polymorphism features



Overview of Frameworks in Android & Java

- Frameworks use Java's inheritance & polymorphism features



See our *Java For Android* MOOC for more info on inheritance & polymorphism

Overview of Frameworks in Android & Java

- Frameworks use Java's inheritance & polymorphism features, e.g.
  - Abstract classes & interfaces provide extension mechanisms

```
public interface Runnable {
  public void run();
}
```

Overview of Frameworks in Android & Java

- Frameworks use Java's inheritance & polymorphism features, e.g.
  - Abstract classes & interfaces provide extension mechanisms
  - Concrete implementations of abstract classes & interfaces are passed to framework

```
public interface Runnable {
  public void run();
}

new Thread(new Runnable() {
  public void run() {
    System.out.println
      ("hello world");
  }
}.start();
```

Anonymous instance of an anonymous inner class

Overview of Frameworks in Android & Java

- Frameworks use Java's inheritance & polymorphism features, e.g.
  - Abstract classes & interfaces provide extension mechanisms
  - Concrete implementations of abstract classes & interfaces are passed to framework

```
public interface Runnable {
  public void run();
}

new Thread(new Runnable() {
  public void run() {
    System.out.println
      ("hello world");
  }
}.start();
```

Anonymous instance of a lambda expression

```
new Thread(() ->
  System.out.println
    ("hello world")).start();
```

See www.drdobbs.com/jvm/lambda-expressions-in-java-8/240166764

End of Common Services & Apps (Part 1): Overview of Object-Oriented Frameworks