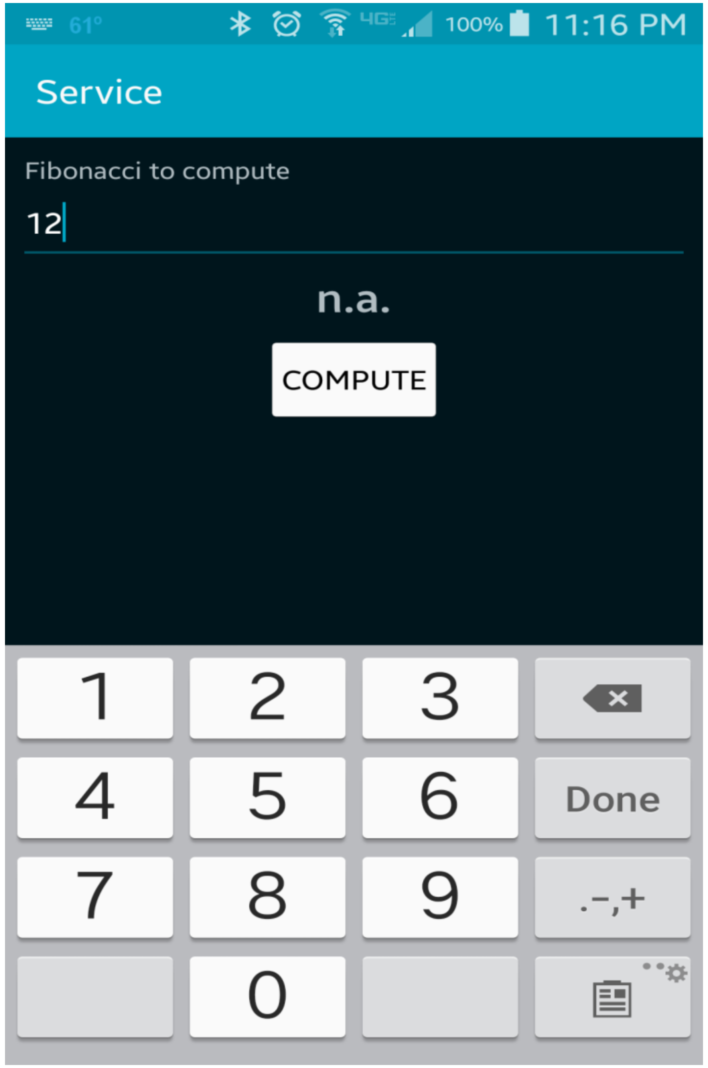
Android – Service (started)

- An Android Service is an application component that can perform long running operations.
- By default, a service runs in the context of the UI thread. I.e., requires a thread to be created for background operations.
- A started service can run indefinitely. It is usually created to perform a single operation that does not return any results (e.g., download a file).
- A started service often posts completion of a task via a notification in the status bar.
- A started service is created via a call to ContextWrapper.startService() (class ContextWrapper is a base class of class Activity).
- A started service needs to derive from class android.app.Service and needs to override onStartCommand().
- A service needs to be declared in AndroidManifest.xml using the <service> tag.
- Helper class IntentService automatically creates a background thread and adds a message queue.



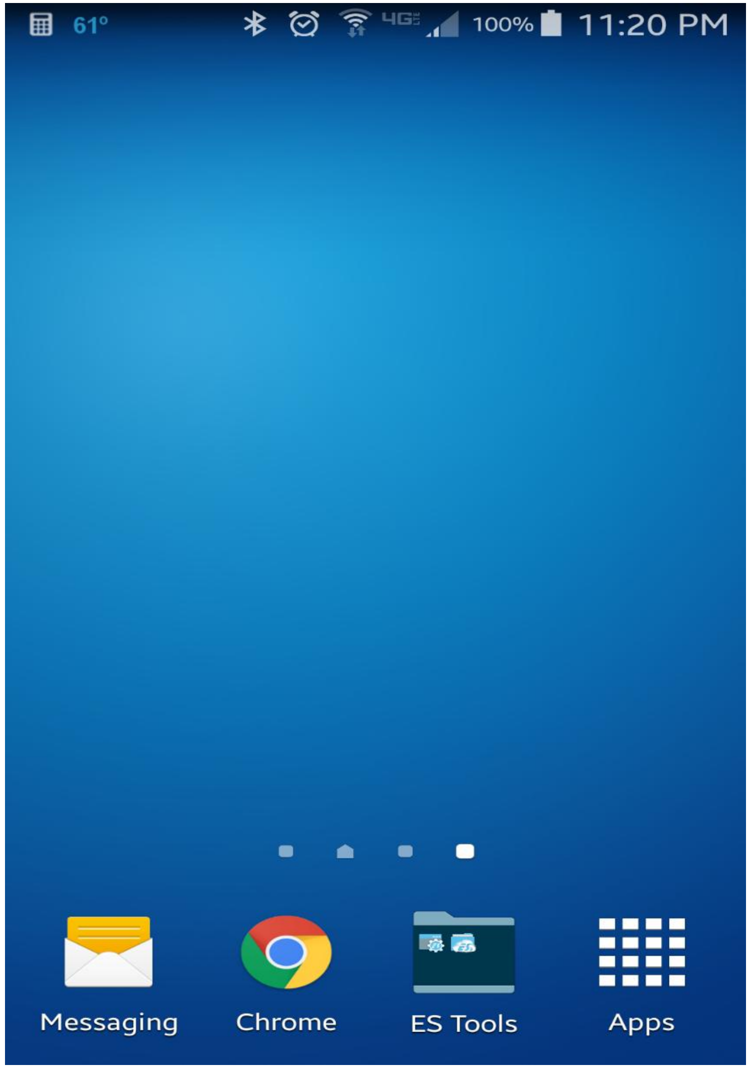
Service

Fibonacci to compute

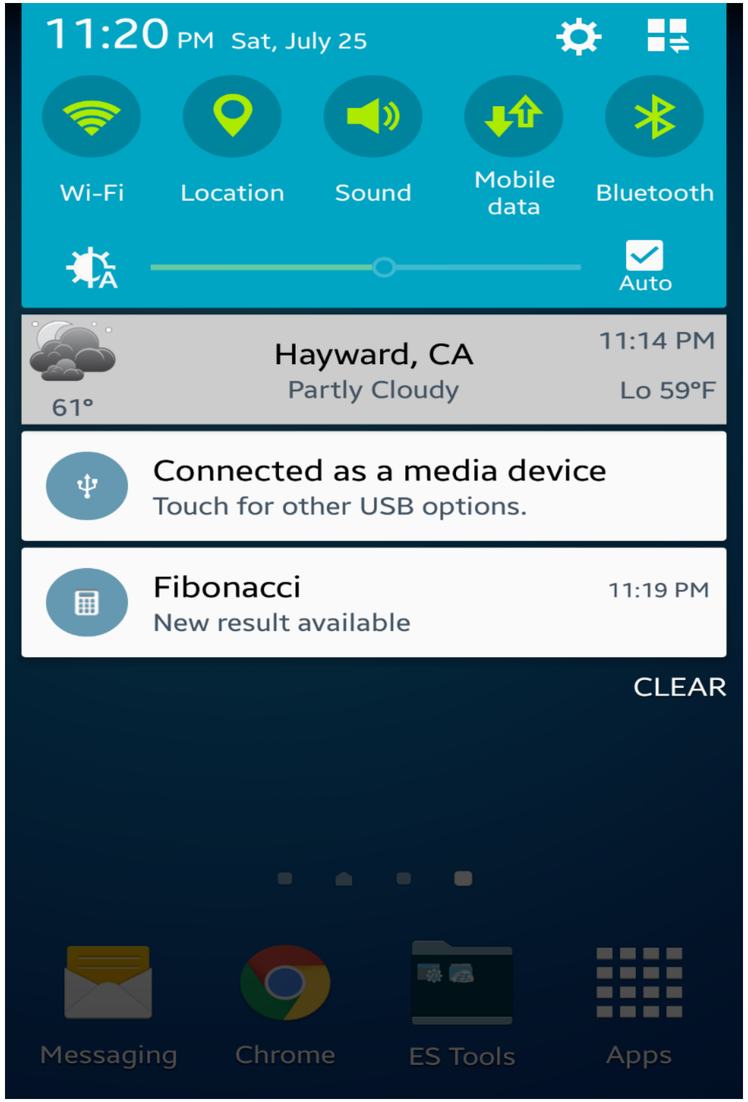
12

n.a.

Step 1: Enter "12" and press "Compute" button. Application will automatically terminate and return to the home screen.



Step 2: After 5 seconds a notification icon will appear in the top left corner of the screen.



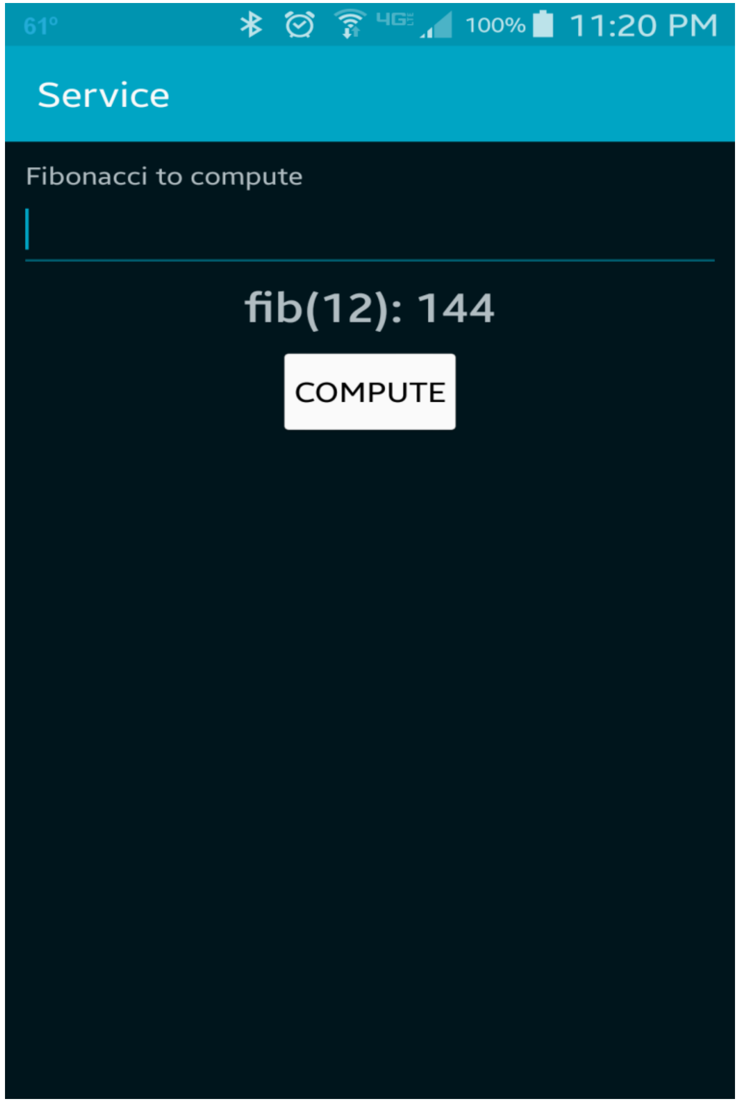
Hayward, CA
Partly Cloudy
61°
11:14 PM
Lo 59°F

Connected as a media device
Touch for other USB options.

Fibonacci 11:19 PM
New result available

CLEAR

Step 3: Swiping from the top of the screen downward will reveal the notifications.



Fibonacci to compute

fib(12): 144

Step 4: Click on the notification will re-launch the app which will then display the result of the computation.

- General overview of Android Services
- android.app.Service
- android.app.IntentService
- android.app.Notification
- android.app.PendingIntent

Exercise

- Change the application such that it will compute the Fibonacci as well as the summation of the input value (i.e., 1 + 2 + 3 + ... + n). The summation should be performed in a separate service. For the new service, derive your implementation from base class android.app.Service (not android.app.IntentService). Use a HandlerThread to run the computation in the background. The summation should force a 3 second delay and post a notification for the result.