Jamarcus Coulter

1. What is task affinity?

The affinity of a task is which activity the task prefers to belongs to. By default, activities of the same app are of the same affinity, but the affinity can be modified. Task affinity in Android generally refers to the different launch modes. Task are pushed onto the task backstack. Task are placed in the stack in the order they start. When a user wants to access an older task, the tasks are popped off the stack until the proper task is found.The task on top of the stack is the one that is running in the foreground. When a new task is brought to the foreground, it is placed on top of the stack and the currently running task is pause and states are saved. The management of these task backstacks is called task affinity. There are four general task affinities:

1. Standard Launch Mode: Allows for multiple tasks of the same app to be saved in the stack.
2. Single Top: Also allows for multiple tasks of the same app to saved in the stack. If a new task of the same app is started while the app is the one on the foreground or top of the stack, the currently running task is removed and the new task is ran. If a task of the same app is already in the stack but not on top, that older task is preserved.
3. Single Instance: Only allows one instance of a task to be in a stack. The system creates a new task and routes the intent to the newly created task.
4. Single Task: Only allows one instance of a task. The system does not create a new task if an instance of the task already exist.

2. How does serialization work?

Serialization in JAVA is the conversion of an object into bitstream to be saved or sent across a medium. To use serialization in JAVA, class must implement the Serializable interface.

3. How does parcels work?

Parcels are the message containers that are passed between processes in Android. Parcel. are sent by the Binder which flattens and reconstructs the parcels. To use parcels, a class must implement the Parcelable interface.

4. What is the difference in an implicit intent and an explicit intent?

Explicit intent is when creating an intent the target component is specified and pass on directly in the intent. With implicit intent, the Android system decides with component the intent will use based on the intent data and actions. The Android system then chooses the component for the intent. If more then one component can handle the intent, a prompt is generally given to the user to decide with component to use. If multiple email apps are installed on the device and the user wants to send an email, a prompt is given to the user on which email app to use is an example of implicit intent.