

1. What's your favorite tool or library for Android? Why is it so useful?

Answer: I have used several libraries in my apps such as volley, gson, picasso, guava, butterknife. But picasso and guava stand out in the lot.

Picasso is a powerful image downloading and caching library. With picasso, image downloading has become very easy. Without picasso, to download a image , one needs to write code for the all the following steps:

- Get Image URL
- Create AsyncTask to download image
- Execute AsyncTask
- Store result in Bitmap
- Set bitmap as source
- Cache image for future

But picasso made this process really easy and the above process can be done in just two steps:

- Get image URL
- Load it into an ImageView with one line:

it also takes care of asynchronous download and caching of image for future use

Guava: It is an open source library provided by google. This library provides many features like collections, range, Objects, Optionals, Strings, Preconditions, in memory caching, Ordering etc. This library makes a developer life easier. This library helps you to reduce the amount of code you need to maintain. This library is highly optimised and provides cleaner code. It also adds function processing capability to java.

2. You want to open a map app from an app that you're building. The address, city, state, and ZIP code are provided by the user. What steps are involved in sending that data to a map app?

1. Get the full address from user.

```
String myAddress = address + city + state + zipCode;
```

2. You need to create a google map URI with address, city, state and zip code as below:

```
Uri gmmIntentUri = Uri.parse("geo:0,0?q=" + myAddress);
```

2. Create an intent and set intent action and data with the above uri.

```
Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
```

3. set intent's package detail to google maps package.

```
mapIntent.setPackage("com.google.android.apps.maps");
```

4. send the intent to the google maps.

```
startActivity(mapIntent);
```

3. Implement a method to perform basic string compression using the counts of repeated characters. For example, the string aabcccccaaa would become a2b1c5a3. If the "compressed" string would not become smaller than the original string, your method should return the original string. The method signature is: "public static String compress(String input)" You must write all code in proper Java, and please include import statements for any libraries you use.

```
import java.util.Scanner;
```

```
public class main {
```

```
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        String enteredString;  
        String compressedString;
```

```

@SuppressWarnings("resource")
Scanner in = new Scanner(System.in);
while(in.hasNext()) {
    enteredString = in.nextLine();
    compressedString = compress(enteredString);
    System.out.print("input string is = " + enteredString + "\n");
    System.out.println("output is " + compressedString + "\n");
}

}

public static String compress(String input) {
    if(!input.equals("") || !input.isEmpty()) {
        int orgLength = input.length();
        if(orgLength <=2) {
            return input;
        }
        StringBuffer bufferString = new StringBuffer();
        char newChar;
        char nextChar;
        for(int i = 0; i<orgLength; i++) {
            newChar = input.charAt(i);
            bufferString.append(newChar);
            int count = 1;
            for(int j = i+1; j<orgLength; j++) {
                nextChar = input.charAt(j);
                if(nextChar == newChar) {
                    count++;
                    if(j == orgLength-1) {
                        bufferString.append(count);
                        i = j;
                        break;
                    }
                }
                }else {
                    bufferString.append(count);
                    i = (j-1);
                    break;
                }
            }
        }
        int compressedLength = bufferString.length();
        if(compressedLength < orgLength) {
            return bufferString.toString();
        }else {
            return input;
        }
    }
    return null;
}
}

```

4. List and explain the differences between four different options you have for saving data while making an Android app. Pick one, and explain (without code) how you would implement it.

Answer: Following are the methods for saving data in android app:

1. Shared Preferences : Used to store private primitive data in key-value pairs. For example: user name and password etc. You can use SharedPreferences class to save any primitive data: booleans, floats, ints, longs, and strings. This data will persist across user sessions

2. Internal Storage : Used to store private data on the device memory. By default, files saved to the internal storage are private to your application and other applications cannot access them. To create and write a private file to the internal storage following functions are used:
openFileOutput(), write(), close().

3. External Storage : Used to store public data on the shared external storage like memory card. This is used to store big files like pdf, image, audio etc. Files saved to the external storage are world-readable and can be modified by the user when they enable USB.

4. SQLite Databases: Used to store structured data in a private database. Data remains on device even after reboots and is removed if data is cleared by user from the settings or the app is uninstalled.

I would like to explain the implementation of Sqlite Database:

1. Define a Schema and Contract class. (The schema is used to create your database. A contract class is used to define names for URIs, tables, and columns)

2. Create a new SQLite database using the SQLiteOpenHelper and override the onCreate() method to execute a SQLite command for creating tables in the database. To write to and read from the database, create an instance of SQLiteOpenHelper using the method getWritableDatabase() and getReadableDatabase(), respectively.

3. implement CRUD (create, read, update and delete) operation on database.

1. Insert information into a Database: To insert data into the database use method insert(). Data can be inserted using ContentValues object to the insert() method.

2. Read information from DataBase: To read data from database, query() method is used by passing the selection criteria and columns. The result of query() is returned in the Cursor object.

3. Delete information from DataBase: To delete rows from database, delete() method is used using the selection criteria.

4. Update information in DataBase: To modify the existing data into the database, update() method is used.

4. Persisting Database Connection: Close the database in the onDestroy() of the calling activity when database is no longer in use.

5. What are your thoughts about Fragments? Do you like or hate them? Why?

Fragments are Android's solution to creating reusable user interfaces. Fragment is part of an activity. Multiple fragment can be combined in a single activity to build a multi-pane UI and reuse a fragment in multiple activities. Fragment has its own life cycle. Fragments can be added and removed while activity is running. Fragment allows user to reuse the code and provides modularity (e.g same view can be used in many activities).

As for liking them, it's more like a love-hate relationship. Their complex lifecycle is one thing I hate. I like to use fragments for multipane interface. Android 4 (ICS) supports both Smartphones and Tablets which means that same application will be running on a smartphone and a tablet. Tablets have bigger screen which will be empty or unused unless it is assigned properly. Fragment make this easier. Fragment can be instantiated either as a whole-screen activity on a phone or as a partial screen on a tablet with only a few lines of code to tell the difference. For example a movie app can use one fragment to show a list of movies on the left and another fragment to display an movie's detail on the right. So both the fragments appear in one activity, side by side. Thus, instead of using one activity to select an movie and another activity to read the movie's detail, the user can select a movie and check its detail within the same activity.

6. If you were to start your Android position today, what would be your goals a year from now?

Mainly I would like to focus on integration, reactive programming and algorithmic skills as it would help me become a better software engineer. I want to build more and more robust and cool apps. I also have plans to contribute towards open source softwares and help the Android community make great products.