What’s your favorite tool or library for Android? Why is it so useful?

I have been using a Picasso for images in Android project and I think it is a very good image loading library. In my opinion, Picasso is very simple to use and its API is easy to understand. So it can reduce the overall time to implement image processing in writing apps. Additionally, it is open sourced and widely used by many Android developers. So I think it is a reliable library.

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?

I think, to open a map app from my app that I'm building. I need to use an implicit intent.

First, I would create a geolocation Uri to hold address info.

Creating a new Intent and passing it Intent.ACTION_VIEW is second.

Then, I'd set the Uri to the Intent using setData().

Finally, I 'd call startActivity() method to launch a map app.

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 StringCompression {
  public static void main(String[] args) {
      System.out.print("Input string: ");
      Scanner scanner = new Scanner(System.in);
      String inputString = scanner.next();
```

```
        System.out.format("Output string: %s", compress(inputString));
        scanner.close();
    }

    public static String compress(String input) {
        // First prepare the string variables and a integer for current char, previous char
        // and count of each compressed character
        String currentChar = "";
        String compressedString = "";
        int countChar = 1;
        // Add first character of the input string into previous string variable
        String previousChar = input.substring(0, 1);

        // Iterate through all of the characters from index position 1
        // because we have already included the character at position 0
        for (int i = 1; i < input.length(); i++) {
            // Compare current character to the character in previous string variable
            currentChar = input.substring(i, i+1);
            // If current character is equal to the character before it,
            // increase value of countChar variable
            if (currentChar.equals(previousChar)) {
                countChar++;
            } else {
                // Otherwise add the previous letter to compressedString variable followed by
                // count of characters
                compressedString = compressedString.concat(previousChar);
                compressedString = compressedString.concat(Integer.toString(countChar));
                // Then set the character counter variable to 1
                // and the previous item to current character
                countChar = 1;
                previousChar = currentChar;
            }
        }

        // Add final character and count to the string to be returned
        compressedString = compressedString.concat(previousChar);
        compressedString = compressedString.concat(Integer.toString(countChar));

        // Check if compressedString is smaller than the input string
        // if compressedString is smaller than the input string then
        // return it otherwise return the input string itself
        if (compressedString.length() < input.length()) {
            return compressedString;
        } else {
            return input;
        }
    }
}
```

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.

1. Internal storage: It is used to save data items on a device. By default, these data are only accessible to the application and removed when the app is uninstalled.

2. External storage: External media storage such as SD cards give the apps an increased amount of storage space, but they require additional considerations. For example, developers cannot assume that a user will have an SD card or other external media resource.

3. Shared preferences: It is used to store simple data items for the app such as integer, boolean, String etc. It stores data as key-value pairs of primitive data types.

4. Databases: You can use an SQLite database to store data using the relational model. There are many advantages to using a database, such as the ability to store and execute structured queries.

Implementing Databases:

1. Create a class that has descriptions of a database.
2. Create a helper class that is going to manage all of the database transactions. This class includes database file and tables structures.
3. Create a database file and all tables.
4. Create a ContentProvider class.
5. Add connectivity between ContentProvider and database so that you can manage data using your ContentProvider.
6. Now, you can use a ContentResolver to query data.

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

When I first learned fragments, I was little confused. It is not easy to understand what fragments are and how to use them. After I have made a few Android apps, I got to know why I should use fragments. Fragments are used like containers of activities. So if I would like to make an Android app which supports multiple devices with different screen sizes, I need to use fragments. It is convenient for the user sometimes to see two different views on the same screen.

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

I would like to be a better Android developer by increasing my skills and knowledge in mobile development. So I would like to focus on analyzing, refactoring and testing skills. And I think teamwork and collaboration are very important qualities for a workplace. I would be looking to improve my communication skills with team members to produce better results.