LAB 1

1.

Description:

* Object: a basic unit of Object-Oriented Programming, represents the real-life entities. An object is a self-contained component that consists of methods and properties to make a particular data useful. An object consists of identity, behavior and state. Identity gives a unique name to an object to make it ability to interact with other objects. State is represented by attributes of an object and behavior is represented by methods of an object. For example, we have an object is dog so the Identity will be the name of the dog; State is age, color or type of the dog and Behavior is the activities of the dog such as sleep, eat or bark. The object is designed as class hierarchies.
* Class: an entity that determines how an object will behave and what the object will contain. We can say that class is a blueprint to build a specific type of object. A class declaration can include:

Modifiers which contains keyword that it is used to set the access level for classes, attributes, methods (example keyword: public, private or default), Class name and Body.

* Instantiation of object: creating new instances of objects to be used in the program
* Visibility(public/private/protected): we have a class declaration include Modifiers which contains keyword that it is used to set the access level for classes, attributes. Corresponding to each of Modifier keyword, there can have different levels of visibility. If the class or method have the keyword is public so it can be seen everywhere including other packages. If the class or method have the keyword is private so it can only be seen and used within this class. If the class or method have the keyword is protected so it can be seen by other classes in the package and sub-class inside or outside the package.
* Member data/methods: block of code which runs when it is called, we can pass the data to the method, because the methods are used to perform actions, so it also known as function
* Inheritance: a mechanism in which an object acquires all the properties and behaviors of a parent object. We can create new classes that are built upon existing classes, reuse their methods and also add the new one.
* Interface: a mechanism to achieve abstraction and multiple inheritance in Java, can only be abstract methods not method body.
* Polymorphism: is a concept by which we can perform a single action in different ways. Polymorphism provides the ability for programmers to call an object’s method, although it has not been determined whether the object has the method or not. Until execution(run-time), the program can identify the object and call the corresponding method of that object.
* Overriding: a feature that allow subclass or child class to provide specific implementation of a method that is already provided by one of its super-classes or parent classes.
* Abstract classes: a class that is declared abstract, may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be subclass.

3.

1. What programming languages you can use for Android app development?

- We can use Kotlin, Java and C++ languages for Android app development.

2. What is .apk file?

- .apk file is an Android package that have the data and resource files which have been compiled by The Android SDK tools. One apk file contains all the contents of an Android app and is the file that Android-powered devices use to install the app.

3. How Android system runs apps?

- Each process has its own virtual machine (VM), so an app's code runs in isolation from other apps. By default, every app runs in its own Linux process. The Android system starts the process when any of the app's components need to be executed, and then shuts down the process when it's no longer needed or when the system must recover memory for other apps.

4. Name four types of Android components. Describe each.

- There are 4 Android components. They are:

+ **Activities:** the entry point for interacting with the user. It represents a single screen with a user interface

+ **Services:** a general-purpose entry point for keeping an app running in the background for all kinds of reasons. It is a component that runs in the background to perform long-running operations or to perform work for remote processes

+ **Broadcast receivers:** a component that enables the system to deliver events to the app outside of a regular user flow, allowing the app to respond to system-wide broadcast announcements. Because broadcast receivers are another well-defined entry into the app, the system can deliver broadcasts even to apps that aren't currently running.

+ **Content providers:** A *content provider* manages a shared set of app data that you can store in the file system, in a SQLite database, on the web, or on any other persistent storage location that your app can access. Through the content provider, other apps can query or modify the data if the content provider allows it

5. What is manifest file and what is its purpose?

- It is the root of android app which user must declare app components like:

+ Identifies any user permissions the app requires, such as Internet access or read access to the user's contacts.

+ Declares the minimum [API Level](https://developer.android.com/guide/topics/manifest/uses-sdk-element#ApiLevels) required by the app, based on which APIs the app uses.

+ Declares hardware and software features used or required by the app, such as a camera, Bluetooth services, or a multitouch screen.

+ Declares API libraries the app needs to be linked against (other than the Android framework APIs), such as the [Google Maps library](http://code.google.com/android/add-ons/google-apis/maps-overview.html).

6. What are resources? Why they are needed?

- An Android app is composed of more than just code—it requires resources that are separate from the source code, such as images, audio files, and anything relating to the visual presentation of the app. For example, you can define animations, menus, styles, colors, and the layout of activity user interfaces with XML files. Using app resources makes it easy to update various characteristics of your app without modifying code. Providing sets of alternative resources enables you to optimize your app for a variety of device configurations, such as different languages and screen sizes

UI Hierarchies

Explain:

The first thing is xml file and is for create a layout for the Main Activity with linear layout, button, text view and edit text

The second thing is java file named Main Activity , it have a string that contain a list of country’s name , next to it , it is an array adapter to push the string into the array and the last is to set the array to list in the layout.