Lab1

1. Study following object oriented concepts

1. Object- An object is an instance of class. It has states and behaviours. For example: A dog has states: colour, name, breed and behaviours: eating, barking etc.

2. Class- Class is like a prototype hat describes the behaviour that supports object of its type.

3. Instantiation of object (creating an object)- The creation of an instance is called an instantiation. Object created from a class.

4. Visibility (public / private / protected)-

Public: It is accessible from other classes too.

Private: It is accessible only in the same class.

Protected: It is accessible only in the same class as well as to the inherited class methods.

5. Member datas / methods- Member datas are variables that belong to an object. A cat object for instance could have member data such as a string color and int age.

6. Inheritance- It means that one class inherits the other class; meaning it has the property of the other class too if it inherits it.

7. Interface- An interface in java is a blueprint of a class. It has static constants and abstract methods. By interface, we can support the functionality of multiple inheritance.

8. Polymorphism- Polymorphism in Java is a concept by which we can perform a single action in different ways

.

9. Overriding- In any object-oriented programming language, Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes.

10.Abstract classes- An abstract class is a class that is declared abstract—it may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be sub-classed.

3. Study Android fundamental concepts

Answer to following questions:

a) What programming languages you can use for Android app development?

Android apps can be written using Kotlin and Java

b) What is .apk file?

Android Package Kit. (.apk) is the package file format used by the Android operating system for distribution and installation of mobile apps.

c) How Android system runs apps?

d) Name four types of Android components. Describe each.

There are four different types of app components which are described below:

1). Activities: An activity is the entry point for interacting with the user. It represents a single screen with a user interface. For example, an email app might have one activity that shoes a list of new emails, another activity to compose and email, and another activity for reading emails.

2) Services: It is a component that runs in the background to perform long-running operations or to perform work for remote processes. For example, a service might play music in the background while the user is in different app, or it might fetch dat over the network without blocking user interaction with an activity

3) Broadcast receivers: A broadcast receiver is 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. For example, an app can schedule an alarm to post a notification to tell the user about an upcoming event and by delivering that alarm to a BroadcastReceiver of the app, there is no need for the app to remain running until the alarm goes off.

4) Content providers: A content provides manages a shared set of app that you can store in the file system, in a SQLite databas, on the web, or on any other persistent storage location the you app can access.

e) What is manifest file and what is its purpose?

Every app project must have an AndroidManifest.xml file at the root of the project source set. The manifest file describes information about your app to the Android build tools, the Android operating system, and Google Play.

If you're using Android Studio to build your app, the manifest file is created for you, and most of the essential manifest elements are added as you build your app.

F) What are resources? Why they are needed?

Resources are the additional files and static content that your code uses, such as bitmaps, layout definitions, user interface strings, animation instructions, and more.