1. Define the following design principles: Singleton, Factory, Builder, Facade, Prototype

Singleton

Singleton is a design pattern about having only one instance of a given class. This pattern is mostly used when there is a unique resource that must be available in the application. For instance, say a piece of hardware, such as a camera or a phone screen. Since there is only one phone screen, it can be encapsulated into a singleton class. This class can act be used to control access to the resource.

To implement a Singleton class in Java. You need a private constructor and a public static method that provides a way for outside classes to use the instance. The method is used both to create the single unique instance in case it hasn’t been created yet, and as a return type that points to the instance.

Factory

Factory design pattern in Object oriented programming focuses on creating objects using superclasses but allowing subclasses to change the type of objects that are created. The pattern allows your code to follow more of the SOLID design principles and loosens the connections between code that focuses on construction and code that uses the objects. With this design pattern, if you need to add a new subclass, you will only need to make the subclass and not make any changes to any existing code.

Facade

A Façade pattern hides the complexities of the system and gives an interface which the client uses to access the system. The interface will provide a simplified access point to the subsystems, allowing a user to be able to get the information they need without needing to know the implementation.

Builder

The builder pattern focuses on separating the construction for a complex object from its representation. This lets us use the same constructor to create objects with different forms. An example would be using an immutable or unchanging class. This class can have data members are optional. Instead of needing individual constructor methods or placing null values into a constructor, it allows for one constructor, which only needs the required information, and having optional variables passed into the builder through its methods, before finally building and initializing the object.

Prototype

Protype pattern is creating new objects by cloning other objects. This is useful when you have groups of classes that are only needed at runtime, while also reducing the need to create subclasses. This is done using cloning in Java. In application, the prototype pattern is useful when your code doesn’t depend the classes from objects that you need to copy, since you would have no way of knowing the object.

1. What is the differences in ART and Dalvik?

Android Runtime uses Ahead of Time compilation, which means that it translates the dex bytecode into machine code and places it in the device’s storage. This occurs only once and allows the code to execute faster. On the other hand, Dalvik uses Just In Time compilation, which means that every time the app is ran, the Dex byte code must be translated to machine code. This impacts the performance of the application, causing it to feel sluggish along with consuming more resources.

1. What is the android manifest used for?

The android manifest provides information about the application to the android system. It describes the components of the applications, being activities, services, broadcast receivers and content providers are and what condition they can launched from. It helps determine what processess will host the application components, and what permissions are needed for the application to interact with other applications. It also declares the minimum level of android API that the application needs to run.

1. How does each of the following units of measure for view work: sp, dp, px, pt, in, mm

sp- Scale independent Pixels

Scale independent pixels are mostly used when specifying font size, as it can be changed by both the screen’s density and the user’s preferred font size.

dp density independent pixels

a unit that is based on the physical density on the screen. The unit is relative to a 160 dpi screen, however the ratio between dp to pixel will change with the screen density. Its mostly used when defining the layout of a UI, as it will ensure proper display of the UI on a screen with a different density.

px-pixels

This is related to the actual pixels on a screen. The units size depends on the screen density.

in- physical inches

inches are density independent. The size of a single in is the same on every screen, however the number of pixels that translates to a physical inch varies.

mm- physical milimeters

this unit is density independent and is the same across all screen densities. It is a measure of a physical millimeter on a screen.

Pt – point

A point is density independent, with 72 pts in an inch. The number of pixels in a pt varies on the screen density.

5. Describe what each section of the Android Platform arch. details.

System Apps

A group of core apps for email, SMS, messaging, calendars, internet browsing. These are default apps included with the system.

Java API Framework

The Java API framework forms the building blocks you need to create android apps by providing a group of modular system components and services, such as:

The View System, which provides a way to build an apps UI

Resource Manager, providing access to non code resources- such as graphics and layout files

Notification Manager, which enables apps to display custome alerts

Activity Manager, which manages the lifecycle of apps

Content Providers, which enable apps to access data from other apps

Native C/C++ Libs

Provides a way to utlize native libraries that use C and C++. If an app requires C or C++ code, you can use the access the Android NDK to access the C++ libraries from your code.

Android Runtime

The android runtime contains the core libraries and Android Runtime

Hardware Abstraction Layer

Provides standard interfaces that expose device hardware capabilities. It contains multiple hardware components, such as the camera or Bluetooth module. When a framework API makes a call to a device, the system loads the library module for that specific component

Linux Kernel

The linux kernel is the foundation for the android platform.

6. Define the following terms: View, ViewGroup, View Hierarchy.

View- A view is an object that is the building blocks UI elements in android. Some of the more common examples of views are EditText, ViewText, Button, ImageView or WebView.

ViewGroup- A viewGroup is a container that can hold both views and other viewGroups. ViewGroups and layouts are interchangeable.

ViewHierachy- A viewHiearchy helps defines the structures of how the view is organized. Each view only has one superview. Its easy to imagine it as a tree, with one viewgroup, being the main container, with other viewgroups and views being contained within it.

7. Explain in detail how the following layouts render, a what unique items each has that must be implemented: Constraint, Linear, Coordinator, Grid and Relative?

ConstraintLayout – if there are no constraints defined, the layout is defined at position (0,0). The top left corner. For constraintLayout, each view must have a minimum of one constraint for each axis

Linear Layout – is a viewgroup that aligns its children either vertically or horizontally. The two unique items for a linear layout is “android::orientation” attribute, which determines whether the children are aligned vertically or horizontally.

CoordinatorLayout – A coordinator layout can create behaviors for specific views its mostly used for two cases, as top level applications or as a container for a child view. These behaviors can be dragging, swiping flinging or any other gesture, making it interactable

Grid – A grid separates the area into cells. Rows and columns are a unique feature of gridLayout. Along with this, Gridlayout can assign a cell for child views if they do not specify a row/column.

Relative – A relative layout is a viewgroup that display its children in relative positions. These positions can be specified to be either relative to the parent view, or to other children in the layout, which is specified by ID