BCA III WEBSOL Android

UNIT – 1

java (Exception handling & Packages & interfaces & JVM & .jar file extension & Multithreading. Database (DML&DDL) , What is Android & Setting up development environment & various editors.

Java is a **programming language** and a **platform**.Java is a high level, robust, object-oriented and secure programming language.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform

Application

According to Sun, 3 billion devices run Java. There are many devices where Java is currently used. Some of them are as follows:

1. Desktop Applications such as acrobat reader, media player, antivirus, etc.
2. Web Applications such as irctc.co.in, javatpoint.com, etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games, etc

Types of Java Applications

There are mainly 4 types of applications that can be created using Java programming:

1) Standalone Application

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine. Examples of standalone application are Media player, antivirus, etc. AWT and Swing are used in Java for creating standalone applications.

2) Web Application

An application that runs on the server side and creates a dynamic page is called a web application. Currently, Servlet, JSP, Struts, Spring, Hibernate, JSF, etc. technologies are used for creating web applications in Java.

3) Enterprise Application

An application that is distributed in nature, such as banking applications, etc. is called enterprise application. It has advantages of the high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.

4) Mobile Application

An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications.

Features of Java

The primary objective of Java programming language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as java *buzzwords*.

A list of most important features of Java language is given below.

1. Simple
2. Object-Oriented
3. Portable
4. Platform independent
5. Secured
6. Robust

Simple

Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun, Java language is a simple programming language because:

- Java syntax is based on C++ (so easier for programmers to learn it after C++).
- Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
- There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

Object-oriented

Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior.

Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Basic concepts of OOPs are:

1. Object

2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

Platform Independent

Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.There are two types of platforms software-based and hardware-based. Java provides a software-based platform.

The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:

1. Runtime Environment
2. API(Application Programming Interface)

Secured

Java is best known for its security. With Java, we can develop virus-free systems.

Robust

Robust simply means strong. Java is robust because:

- It uses strong memory management.
- There is a lack of pointers that avoids security problems.
- There is automatic garbage collection in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
- There are exception handling and the type checking mechanism in Java. All these points make Java robust.

Portable

Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

High-performance

Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++). Java is an interpreted language that is why it is slower than compiled languages, e.g., C, C++, etc.

Exception Handling

The **Exception Handling in Java** is one of the powerful *mechanism to handle the runtime errors* so that normal flow of the application can be maintained.

In this page, we will learn about Java exceptions, its type and the difference between checked and unchecked exceptions.In Java, an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime

Advantage of Exception Handling

The core advantage of exception handling is **to maintain the normal flow of the application**. An exception normally disrupts the normal flow of the application that is why we use exception handling.

Types of Java Exceptions

There are mainly two types of exceptions: checked and unchecked. Here, an error is considered as the unchecked exception. According to Oracle, there are three types of exceptions:

1. Checked Exception
2. Unchecked Exception
3. Error

Difference between Checked and Unchecked Exceptions

1) Checked Exception

The classes which directly inherit Throwable class except RuntimeException and Error are known as checked exceptions e.g. IOException, SQLException etc. Checked exceptions are checked at compile-time.

2) Unchecked Exception

The classes which inherit RuntimeException are known as unchecked exceptions e.g. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime.

3) Error

Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc

Java Exception Keywords

There are 5 keywords which are used in handling exceptions in Java.

Keyword	Description
try	The "try" keyword is used to specify a block where we should place exception code. The try block must be followed by either catch or finally. It means, we can't use try block alone.
catch	The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.
finally	The "finally" block is used to execute the important code of the program. It is executed whether an exception is handled or not.
throw	The "throw" keyword is used to throw an exception.
throws	The "throws" keyword is used to declare exceptions. It doesn't throw an exception. It specifies that there may occur an exception in the method. It is always used with method signature.

```
public class JavaExceptionExample{
  public static void main(String args[]){
   try{
      //code that may raise exception
      int data=100/0;
   }catch(ArithmeticException e){System.out.println(e);}
   //rest code of the program
   System.out.println("rest of the code...");
  }
}
```

```
Exception in thread main
java.lang.ArithmeticException:/ by
zero
rest of the code...
```

Common Scenarios of Java Exceptions

There are given some scenarios where unchecked exceptions may occur. They are as follows:

1) A scenario where ArithmeticException occurs

If we divide any number by zero, there occurs an ArithmeticException.

```
int a=50/0;//ArithmeticException
```

2) A scenario where NullPointerException occurs

If we have a null value in any variable, performing any operation on the variable throws a NullPointerException.

```
String s=null;
System.out.println(s.length());//NullPointerException
```

3) A scenario where NumberFormatException occurs

The wrong formatting of any value may occur NumberFormatException. Suppose I have a string variable that has characters, converting this variable into digit will occur NumberFormatException.

```
String s="abc";
int i=Integer.parseInt(s);//NumberFormatException
```

---

4) A scenario where ArrayIndexOutOfBoundsException occurs

If you are inserting any value in the wrong index, it would result in ArrayIndexOutOfBoundsException as shown below:

```
int a[]=new int[5];
a[10]=50; //ArrayIndexOutOfBoundsException
```

Try block

Java **try** block is used to enclose the code that might throw an exception. It must be used within the method.If an exception occurs at the particular statement of try block, the rest of the block code will not execute. So, it is recommended not to keeping the code in try block that will not throw an exception.

Java try block must be followed by either catch or finally block.

catch block

Java catch block is used to handle the Exception by declaring the type of exception within the parameter. The declared exception must be the parent class exception (i.e., Exception) or the generated exception type. However, the good approach is to declare the generated type of exception.

java package

A **java package** is a group of similar types of classes, interfaces and sub-packages.Package in java can be categorized in two form, built-in package and user-defined package.There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.Here, we will have the detailed learning of creating and using user-defined packages.

Advantage of Java Package

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained

2) Java package provides access protection.

3) Java package removes naming collision.

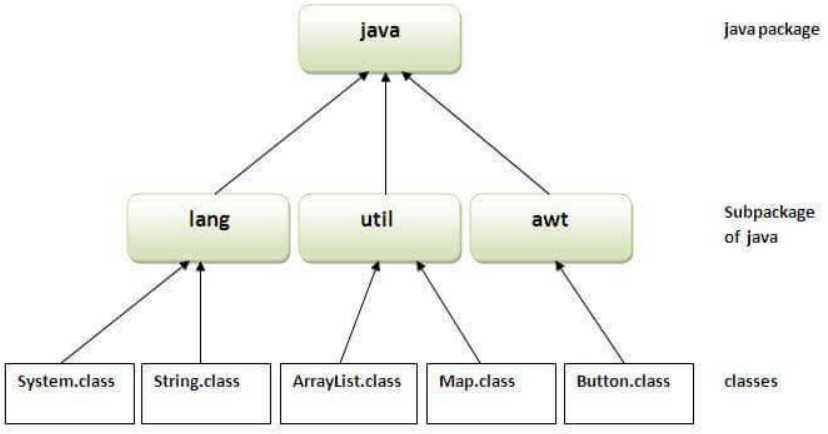
The **package keyword** is used to create a package in java

How to compile java package

If you are not using any IDE, you need to follow the **syntax** given below:

1. javac -d directory javafilename

The -d switch specifies the destination where to put the generated class file. You can use any directory name like /home (in case of Linux), d:/abc (in case of windows) etc. If you want to keep the package within the same directory, you can use . (dot).



How to access package from another package?

There are three ways to access the package from outside the package.

1. import package.*;
2. import package.classname;
3. fully qualified name.

*1) Using packagename.**

If you use package.* then all the classes and interfaces of this package will be accessible but not subpackages.

The import keyword is used to make the classes and interface of another package accessible to the current package.

2) Using packagename.classname

If you import package.classname then only declared class of this package will be accessible.

3) Using fully qualified name

If you use fully qualified name then only declared class of this package will be accessible. Now there is no need to import. But you need to use fully qualified name every time when you are accessing the class or interface.

interface

An **interface in java** is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is *a mechanism to achieve abstraction*. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.

In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body.

Why use Java interface?

There are mainly three reasons to use interface. They are given below.

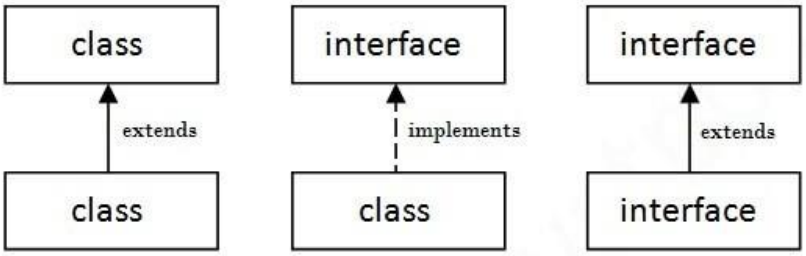
- It is used to achieve abstraction.
- By interface, we can support the functionality of multiple inheritance.
- It can be used to achieve loose coupling.

How to declare an interface?

An interface is declared by using the interface keyword. It provides total abstraction; means all the methods in an interface are declared with the empty body, and all the fields are public, static and final by default. A class that implements an interface must implement all the methods declared in the interface.

The relationship between classes and interfaces

As shown in the figure given below, a class extends another class, an interface extends another interface, but a **class implements an interface**.



JVM

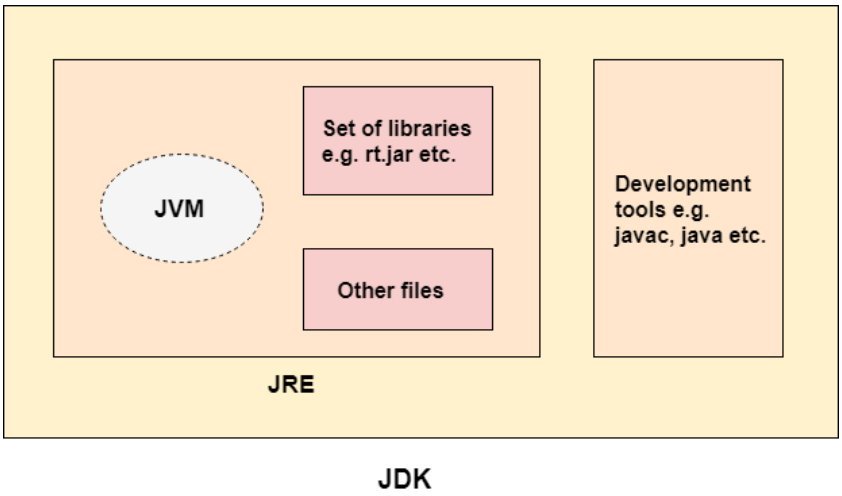
JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.JVMs are available for many hardware and software platforms. JVM, JRE, and JDK are platform dependent because the configuration of each OS is different from each other. However, Java is platform independent. There are three notions of the JVM: *specification*, *implementation*, and *instance*.

The JVM performs the following main tasks:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

JVM provides definitions for the:

- Memory area
- Class file format
- Register set
- Garbage-collected heap
- Fatal error reporting etc.



JAR

A JAR (Java Archive) is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file to distribute application software or libraries on the Java platform.
In simple words, a JAR file is a file that contains compressed version of .class files, audio files, image files or directories. We can imagine a .jar files as a zipped file(.zip) that is created by using WinZip software. Even , WinZip software can be used to used to extract the contents of a .jar . So you can use them for tasks such as lossless data compression, archiving, decompression, and archive unpacking.

Multithreading

Multithreading in java is a process of executing multiple threads simultaneously.

A thread is a lightweight sub-process, the smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.

However, we use multithreading than multiprocessing because threads use a shared memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.Java Multithreading is mostly used in games, animation, etc.

Advantages of Java Multithreading

1) It **doesn't block the user** because threads are independent and you can perform multiple operations at the same time.

2) You **can perform many operations together, so it saves time**.

3) Threads are **independent**, so it doesn't affect other threads if an exception occurs in a single thread.

What is Thread in java

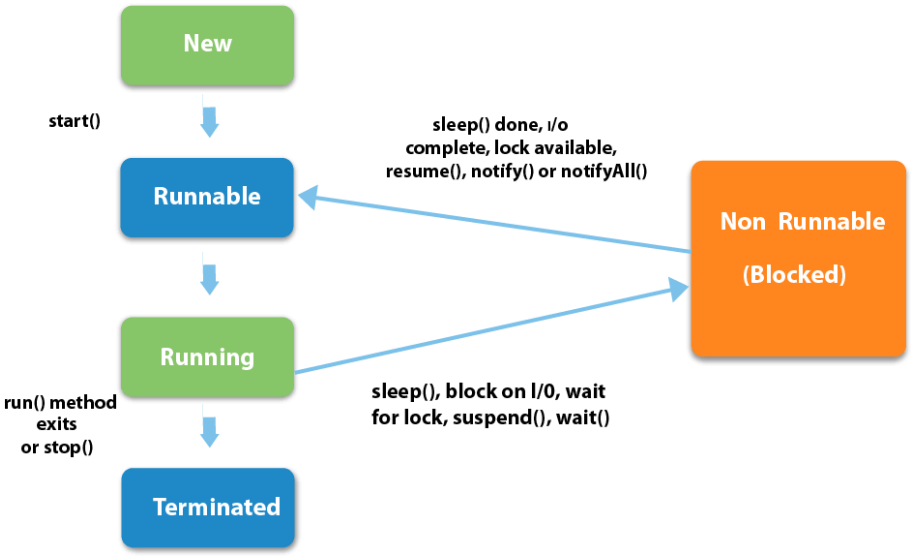
A thread is a lightweight subprocess, the smallest unit of processing. It is a separate path of execution.Threads are independent. If there occurs exception in one thread, it doesn't affect other threads. It uses a shared memory area.

thread life cycle

A thread can be in one of the five states. According to sun, there is only 4 states in **thread life cycle in java** new, runnable, non-runnable and terminated. There is no running state.But for better understanding the threads, we are explaining it in the 5 states.

The life cycle of the thread in java is controlled by JVM. The java thread states are as follows:

1. New
2. Runnable
3. Running
4. Non-Runnable (Blocked)
5. Terminated



1) New

The thread is in new state if you create an instance of Thread class but before the invocation of start() method.

2) Runnable

The thread is in runnable state after invocation of start() method, but the thread scheduler has not selected it to be the running thread.

3) Running

The thread is in running state if the thread scheduler has selected it.

4) Non-Runnable (Blocked)

This is the state when the thread is still alive, but is currently not eligible to run.

5) Terminated

A thread is in terminated or dead state when its run() method exits.

How to create thread

There are two ways to create a thread:

1. By extending Thread class
2. By implementing Runnable interface.

1) Java Thread Example by extending Thread class

```
class Multi extends Thread{
public void run(){
System.out.println("thread is running...");
}
public static void main(String args[]){
Multi t1=new Multi();
t1.start();
 }
}
```

```
Output:thread is running...
```

---

2) Java Thread Example by implementing Runnable interface

```
class Multi3 implements Runnable{
public void run(){
System.out.println("thread is running...");
}

public static void main(String args[]){
Multi3 m1=new Multi3();
Thread t1 =new Thread(m1);
t1.start();
 }
}
```

```
Output:thread is running...
```

Database

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

For example: The college Database organizes the data about the admin, staff, students and faculty etc.

Using the database, you can easily retrieve, insert, and delete the information.

Database Management System

- Database management system is a software which is used to manage the database. For example: MySQL, Oracle, etc are a very popular commercial database which is used in different applications.
- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
- It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

DBMS allows users the following tasks:

- **Data Definition:** It is used for creation, modification, and removal of definition that defines the organization of data in the database.
- **Data Updation:** It is used for the insertion, modification, and deletion of the actual data in the database.
- **Data Retrieval:** It is used to retrieve the data from the database which can be used by applications for various purposes.
- **User Administration:** It is used for registering and monitoring users, maintain data integrity, enforcing data security, dealing with concurrency control, monitoring performance and recovering information corrupted by unexpected failure.

Characteristics of DBMS

- It uses a digital repository established on a server to store and manage the information.
- It can provide a clear and logical view of the process that manipulates data.
- DBMS contains automatic backup and recovery procedures.
- It contains ACID properties which maintain data in a healthy state in case of failure.
- It can reduce the complex relationship between data.
- It is used to support manipulation and processing of data.
- It is used to provide security of data.
- It can view the database from different viewpoints according to the requirements of the user.

Advantages of DBMS

- **Controls database redundancy:** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.
- **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.
- **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
- **Reduce time:** It reduces development time and maintenance need.
- **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
- **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

Disadvantages of DBMS

- **Cost of Hardware and Software:** It requires a high speed of data processor and large memory size to run DBMS software.
- **Size:** It occupies a large space of disks and large memory to run them efficiently.

- **Complexity:** Database system creates additional complexity and requirements.
- **Higher impact of failure:** Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

SQL

- SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).
- It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.
- All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language.
- SQL allows users to query the database in a number of ways, using English-like statements.

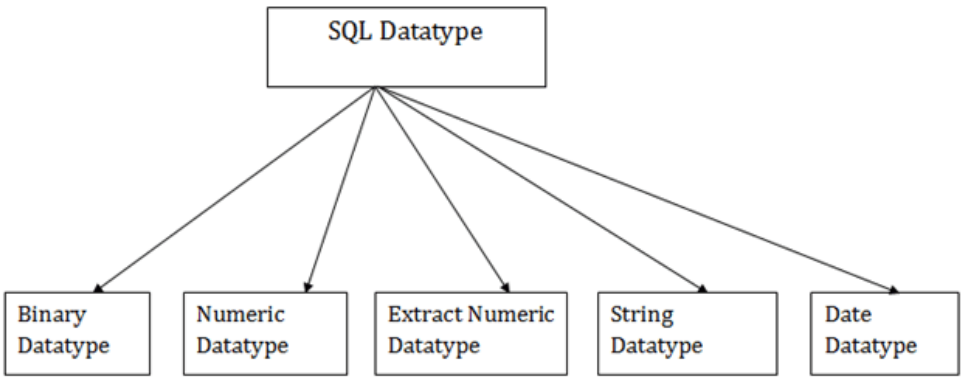
Characteristics of SQL

- SQL is easy to learn.
- SQL is used to access data from relational database management systems.
- SQL can execute queries against the database.
- SQL is used to describe the data.
- SQL is used to define the data in the database and manipulate it when needed.
- SQL is used to create and drop the database and table.
- SQL is used to create a view, stored procedure, function in a database.
- SQL allows users to set permissions on tables, procedures, and views.

Datatype

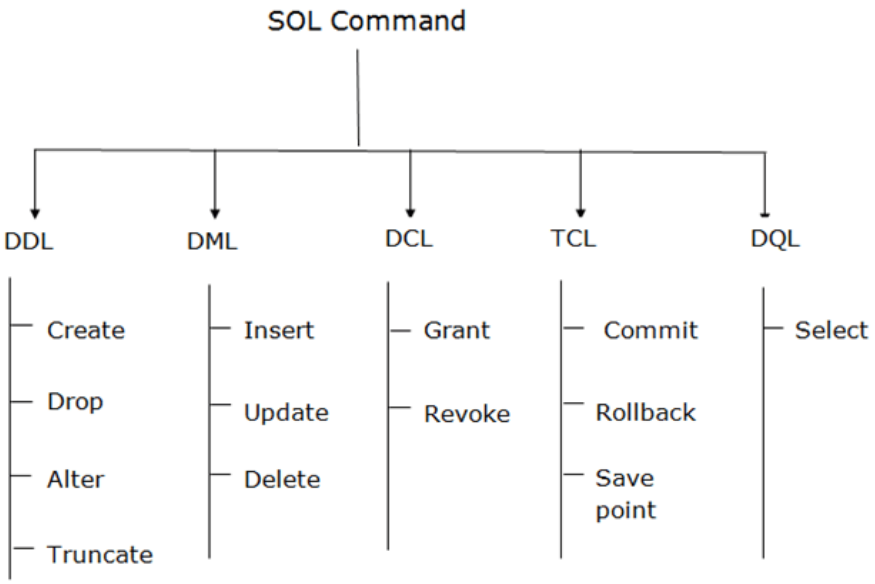
- SQL Datatype is used to define the values that a column can contain.
- Every column is required to have a name and data type in the database table.

Datatype of SQL:



command

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.



Data definition language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

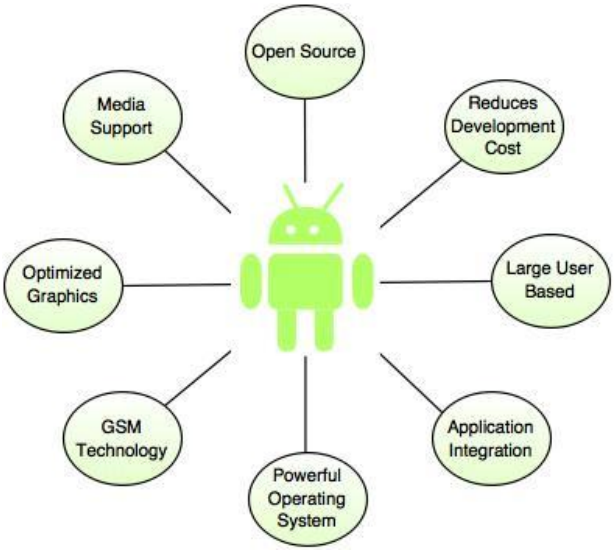
- INSERT

- UPDATE
- DELETE

What is Android

Android is an operating system which is based on the Linux kernel. Android system is also called as **Android Open Source Project (AOSP), lead by Google**. Android is used for mobile devices, such as smart phones and tablet computers. The Android application makes life more comfortable and advanced for the users.

Features of Android



The above diagram shows some features of Android:

- Android is a powerful, open source, Linux – based operating system.
- It provides a rich development environment for building the applications.
- Android has a higher success ratio, because it reduces the development cost.
- Android has a large developer community which integrates the internal application.
- It has reached the top of the smart phone market segment and day by day its user base is growing strong.
- Android supports audio, video formats like JPEG, PNG, GIF, BMP, MP3, MP4, MIDI, AMR, AMR-WB, MPEG-4 etc.

History of Android

- Android was developed by the **Open Handset Alliance, led by Goggle.**
- The first version of Android was released by Google in 2007 and the Android version 1.0 was released in September 2008.

- In June 2012, Google announced the next Android version 4.1 Jelly Bean. This version of Android is an incremental update which is mostly used for improving the user interface.

Advantages of Android

- Android is largely supported by Google allowing you to use various services of Google.
- Android is an open source and runs on mobile devices, tablets etc.
- It is multitasking that means you can run many applications at the same time. For example, you can browse Facebook while listening the song.
- The Android operating system is available on mobile phones from various manufacturers like Samsung, Motorola, HTC, Sony Ericsson etc.
- Using Android phone, you can easily check e-mail from Gmail if your Gmail account is integrated with Google Services.
- User can easily access a variety of settings quickly and easily.

Disadvantages of Android

- Android requires continuous Internet connection if you are using Google services.
- Android shows error & forces to close the large apps/games, which is very annoying.
- It takes large amount of mobile data if a large number of background processes are running.
- It increases the usage of RAM and decreases battery performance when many processes are running in the background.

Android Application Components

- Application components are the essential part of the Android operating system which is used to build the Android applications.
- The *AndroidManifest.xml* file describes each component of the application and how they interact.

Following are the Android Application Components,

1. Activities
2. Services
3. Content Providers
4. Broadcast Receivers
5. Intents

1. Activities

- Activity is also known as Widgets.

- Activity represents a single screen with a user interface.
- It is an individual user interface screen in an Android Application where visual elements called Views.
- It interacts with the user to do only one thing, such as unlock screen, dial a phone, etc.
- If new activity starts, then previous activity is stopped, but the data is preserved.
- An application consists of multiple activities.

  For example, an email application has one activity to display a list of new emails, another activity is to compose email, reading email and so on.

2. Services

- Services performs the action without user interaction in the background, but does not get initiated without user invocation.
- It does not require a user interface.
- It is an android application component which runs in a background and has no visual UI.
- It is used to perform the processing part of your application in the background.

  For example, music player application. When the music station is playing the song, the user can open another application and the song plays in the background.

3. Content Providers

- Content providers are the android application component that provide a flexible way to make data available across applications.
- It manages common data based on permissions.
- It manages the data which is being shared by more than one application.
- Content provider is a critical concept that has led to develop in-house android applications in a better way.

4. Broadcast Receivers

- Broadcast receivers are used to receive messages which are broadcast by the Android applications.

  For example, the warning where the battery is getting low, change of time zone, etc.
- They respond to broadcast messages from other applications.

5. Intent

- Intent is a message which allows components to request activities from other components.

  There are two types of Intents,

  1. Explicit Intents
  2. Implicit Intents

  1. **Explicit intents** are used for application's internal communications.

  2. **Implicit intent** means sending a message to the Android system to find a suitable activity that can be responded to the intent.

Android architecture

Android architecture is a stack of software components. It is in the form of a software application, operating system, run-time environment, middleware, native libraries and services.

It is categorized into five parts as below:

1. Linux Kernel
2. Native Libraries
3. Android Runtime
4. Application Framework
5. Applications

Each part of the stack and the elements within each layer are integrated and provide optimal application development and execution environment for mobile devices.

The following diagram shows the architecture of Android,

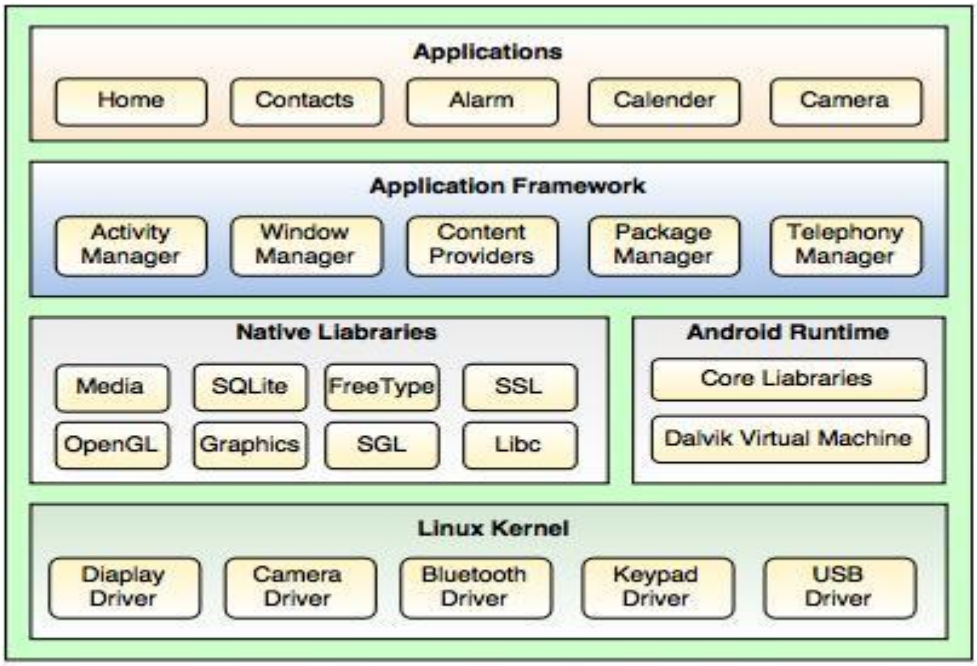


Fig. Android Architecture

1. Linux Kernel

- Linux is the heart of Android architecture.
- It provides a level of abstraction between the hardware devices and the upper layers of the Android software stack.
- The Android operating system is based on the Linux kernel.

- The Linux kernel is responsible for various device drivers such as Camera driver, Display driver, Bluetooth driver, Keypad driver, Memory management, Process management, Power management, etc.

2. Native Libraries

- The native libraries such as Media, WebKit, SQLite, OpenGL, FreeType, C Runtime library (libc) etc. are situated on the top of a Linux kernel.
- Media library is responsible for playing and recording audio and video formats, FreeType is for font support, WebKit is for browser support, SQLite is for database, SSL is for Internet security etc.

3. Android Runtime

- Android Runtime is the third section of the architecture and situated on the second layer from the bottom.
- Android Runtime includes core libraries and Dalvik Virtual Machine (DVM) which is responsible to run android application.
- Dalvik Virtual Machine (**DVM**) is like Java Virtual Machine (**JVM**) in Java, but DVM is optimized for mobile Devices.
- DVM makes use of the Linux core features like memory management and multi-threading, which are essential in the Java language.
- DVM provides fast performance and consumes less memory.

4. Application Framework

- Application framework is situated on the top of the Native libraries and Android runtime.
- Android framework provides a lot of classes and interfaces for Android application development and higher level services to the applications in the form of Java classes.
- It includes Android API's such as Activity manager, Window manager, Content Provider, Telephony Manager, etc.
- Activity manger is responsible for controlling all the aspects of the application lifecycle and activity stack, Content provider is responsible for allowing the applications to publish and share the data with the other applications, View system is responsible for creating application user interfaces, etc.

5. Applications

- Applications are situated on the top of the Application framework.
- The applications such as Home, Contact, Alarm, Calender, Camera, Browsers, etc. use the Android framework which uses Android runtime and libraries. Android runtime and Native libraries use Linux kernel.
- The user can write his/her application to be installed on this layer only.

Categories of Android applications

There are many android applications in the market. The top categories are:

- Entertainment
- Tools
- Communication
- Productivity
- Personalization
- Music and Audio
- Social
- Media and Video
- Travel and Local etc.

Android Development Environment Setup

Generally to build an applications for Android we should have Java Development Kit (JDK), Android SDK, and a development environment.
The Android SDK is compatible with Windows, Mac and Linux operating systems to build android applications based on our requirements.
We can setup android development environment using following two ways

1. Setup Eclipse IDE Manually (**Depreciated**)
2. Android Studio

Initially Google supported a Manual **Eclipse IDE Setup** for android development environment by downloading required components like Eclipse IDE, Android SDK, Java Development Kit (JDK) etc. from official site. Afterwards Google introduced a component called **Android Studio** to make environment setup process simple.
By using **Android Studio** bundle we can easily setup android development environment in any operating system to implement android applications.

Android Studio is the combination of following components to allow users to implement android applications.

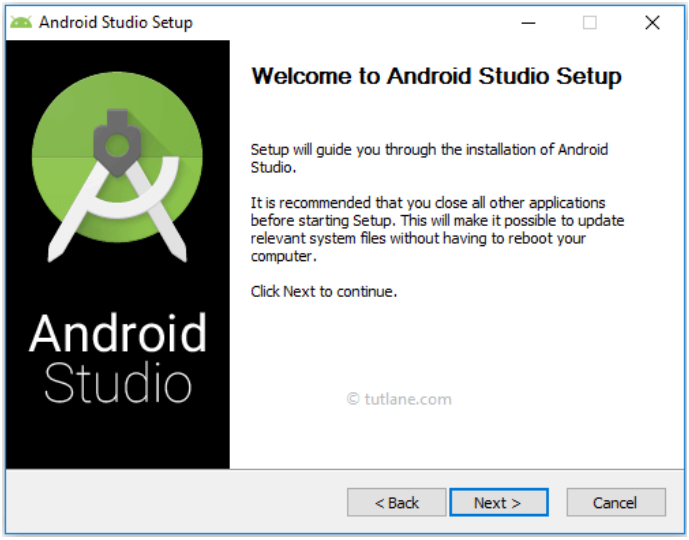
1. Eclipse IDE
2. Android SDK
3. Android Virtual Device
4. Eclipse Plugin

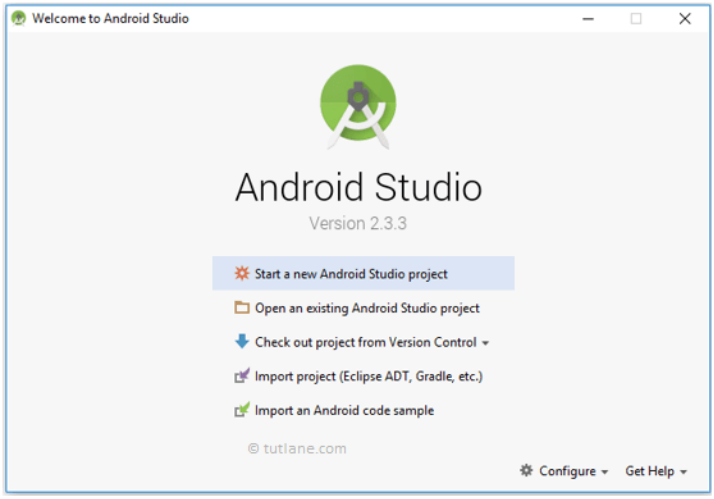
By downloading **Android Studio** directly from Google website to setup we can easily setup development environment.Initially to setup an android development environment in our system we need to install following components manually by downloading from different sites.

1. Eclipse IDE
2. Eclipse Plugin

3. Android SDK

To make android development environment setup process simple Google introduced a new android IDE called **Android Studio**. The **Android Studio** will contain all the required components like Eclipse IDE, Eclipse Plugin and Android SDK so we don't need to download the components separately.





After completion of all required components installation we will be able to see Android Studio welcome window like as shown below.

Android Layout File (activity_main.xml)

The UI of our application will be designed in this file and it will contain **Design** and **Text** modes. It will exists in **layouts** folder and the structure of **activity_main.xml** file in **Design** mode

Android Main Activity File (MainActivity.java)

The main activity file in android application is **MainActivity.java** and it will exists in **java** folder.
The **MainActivity.java** file will contain the java code to handle all the activities related to our app.

Android Manifest File (AndroidManifest.xml)

Generally our application will contain multiple **activities** and we need define all those **activities** in **AndroidManifest.xml** file. In our manifest file we need to mention the main activity for our app using **MAIN** action and **LAUNCHER** category attributes in **intent filters** (<intent-filter>). In case if we didn't mention MAIN action or LAUNCHER category for main activity, our app icon will not appear in home screen's list of apps.