DON Basco Institute of Technology, Kurla, Mumbai

Date:- 29/11/21 Branch:- IT Sem:- 7 Seat:- 12277937
Day:- Mon Sub:- MAD Sign:- Byte
Time:- 3-5.30pm Page:- 1 / 19

Q SON

1 1 Which of the following is NOT an important folder included withing an apk file
→ (D) Test/

2. Java byte code are converted into dalvik byte code using ... program.
→ (D) Dex compiler

3. Inflating is a process of ...
→ (C) Loading an application using layout file

4. What command is used to obtain descriptions of all the permissions defined on device
→ (B) adb shell pm list permission -s

5. Android uses gradle for
→ (A) Packing & compiling android apps

Date:- 29/11/24 Branch:- IT Sem:- 7 Seat:- 122177933
Day:- Mon Sub:- MAD Sign:- Gupta
Time:- 3-5.30pm Page:- 2/19

Q3 (A)

6 Which is the standard prefix used for query string URI by content provider?
→ (A) content://

7 In Android, SQLite by default saves data in
→ (C) Internal storage

8. Which tool is used to extract the Android Manifest file from APK file?
→ (C) Apktool

9. During an activity lifecycle, what is the callback method used to start an activity from stop state?
→ (B) onRestart()

10 What does this code do?.

```
Intent = new Intent()
intent.setAction(Intent.ACTION_VIEW);
intent.setData(Android.net.Uri.parse("https://www.
        androidder.org"));
startActivity(Intent)
```

→ (B) Starts an activity using implicit intent

Don Bosco Institute of Technology, Kurla, Mumbai

Date:- 29/11/21 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:- Gupta
Time:- 3-5.30 Page:- 3/19

Q	SQN
3	A

Difference between:-
(a) Java virtual machine and Dalvik virtual Machine.

Java Virtual Machine	Dalvik Virtual Machine
- Stack based VM that performs arithmetic and logic operations through push and pop operands. The result of operations is stored in stack memory	- Register based VM that uses registers located in the CPU to perform arithmetic & logic operations
- Java source code is compiled into Java bytecode format (.classfile) that further translates into machine code	- Source code files that are first of all compiled into Java bytecode format like JVM. Further, the DEX compiler (dx tool) converts the java byte code into Dalvik byte code (classes.dex) file that will be used to create .apk file.
- More information is required to the VM for data loading and manipulation as well as loading is stack data structure	Instruction size is larger as it needs to encode the source and destination register of VM.

Don Bosco Institute of Technology, Kurla, Mumbai

Date:- 29/11/21 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:- Gupta
Time:- 3-5:30 Page:- 4 / 19

Q | Soln

Java Virtual Machine	Dalvik Virtual Machine
- Compiled bytecode size is compact because the location of operand is implicity on the operand stack	Compiled bytecode size is larger as each instruction needs all implicit operations.
- Executable file is .jar	Executable file for the device is .apk file
- Supports multiple operating systems like windows, Linux, & MacOS	Supports only the Android operating system.

(b)
Activity and Fragment

Activity	Fragment
- Activity is an application component that gives a user interface where the user can interact	The fragment is only part of an activity. It basically contributes its UI to an activity
- Activity is not dependent on fragment	Fragment is dependent on activity & can't exist independently

Don Bosco Institute of Technology, Kurla, Mumbai

Date:- 29/11/24 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:- Gupta
Time:- 3-5-30 Page:- 5/ 19

Q | SOL

Activity	Fragment
- We need to mention all activity in manifest.xml file	Fragment is not required to mention in manifest.xml file.
- We can't create multi-screen UI without using fragment in an activity	After using multiple fragments in a single activity, we can create multi screen UI.
- Activity can exist without a fragment	Fragment cannot be used without activity.
- Activity is not light weight	Fragment is light weight.
- Lifecycle methods are hosted by OS. The activity has its own life cycle.	Lifecycle methods in fragments are hosted by hosting the activity.

Don Bosco Institute of Technology, Kurla, Mumbai

Date:- 29/11/21 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:- Gupta
Time:- 3-5:30 Page:- 6/19

Q	SQN
3	B

The following ways are some of the best practices that must be followed while implementing security in android apps:-

Enforce secure communication :-
When you safeguard the data that you exchange between your app and other apps, stability of app is improved and the data seat is protected.
It can be done by:-

(i) Use of implicit intentents and non exported content providers
⇒ Show an app chooser
If an implicit intent can launch at least two possible apps, show both so user can explicitly select the app they trust

(ii) Apply signature based permissions
These permissions don't require user confirmation and instead check that the apps accessing the data are signed using same key.

(iii) Disallow access to your app's content providers
You should explicitly disallow other developer's apps from accessing content Provider objects

Don Bosco Institute of Technology, Kurla, Mumbai

Date:- 29/11/24 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:- Gupta
Time:- 3-5:30 Page:- 7/19

9 Soln

(iv) Ask for credentials before showing sensitive information:-
When requesting credentials, ask for either a PIN/password/pattern or a biometric credential

(v) Apply network security measures
This can be done by:-
(i) Use SSL traffic:-
If your app communicates with a web server, use HTTPS request

(ii) Add a network security configuration:-
If your app uses new or custom CAs, you can declare that in a configuration file

(iii) Use WebViews Objects carefully
(iv) Use HTML message channels

Provide right permissions
This can be done by:-
(i) Use intents to defer permissions
Whenever possible, don't add an permission to your app that could be completed by another app, instead defer request.

Steed (ii) Share data securely across apps

Don Bosco Institute of Technology, Kurla, Mumbai

Date:- 29/11/24 Branch:- IT Sem:- 7 Seat:- 12277933

Day:- Mon Sub:- MAD Sign:- dpb

Time:- 3-5:30 Page: 8/19

Q sol

Store data safely

This can be done by:-

(i) Store sensitive data within internal storage

(ii) Store data in external storage based on use case

(iii) Check availability of storage volume

(iv) Access app specific files

(v) Check validity of data

(vi) Store only non-sensitive data in cache

(vii) Use Shared Preferences in private mode

Keep services and dependencies up-to-date.

Don Bosco Institute of Technology, Mumbai

Date:- 29/11/21 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:-
Time:- 3-5:30 Page:- 9/19

Q SQN
:= 2 C For developing location aware application in android, it needs location providers, There are two types of location providers
(1) GPS Location Provider
(2) Network location Provider

Steps to get location in Android
(1) Provide permissions in manifest file for receiving location update
(2) Create location manager instance as reference to location service
(3) Request location from LocationManager
(4) Receive location update from LocationListner on change of location

Step1:- Provide permssion for receiving update

```
<manifest..>
    <uses-permission android:name = "android.
    permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name = "android.
    permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name = "android
    permission.INTERNET" />
</manifest
```

ACCESS_COARSE_LOCATION is used when we use network location provider.
ACCESS_FINE_LOCATION works for both providers
INTERNET is must for network provider.

Date:- 29/11/21 Branch:- IT sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:- Gupta
Time:- 3-5.30 Page:- 16/19

SON

Step 2:- Create LocationManager instance as reference to location service

For any background android service, we need to get reference for using it. similarly, location service reference will be created by using getSystemService()

```
LocationManager = (LocationManager)
                  getSystemService (Context.
                  LOCATION_SERVICE);
```

Step 3:- Request current location from LocationManager

After creating location service reference, location updates are requested using requestLocationUpdates() method.
This function requires type of location provider, no. of seconds, distance & LocationListener object over which location is to be updated.

```
locationManager.requestLocationUpdates(
LocationManager.GPS_Provider, 0, 0, this)
```

(4) Step 4:- Request location update from LocationListener will be notified based on interval specified or no. of seconds

Don Bosco Institute of Technology, Kurla, Mumbai

Date:- 29/11/21 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:- Supta
Time:- 3-5:30 Page:- 11/19

Q SQN
2 B.

Service Component -

- A service is a general-purpose entry point for keeping an app running in background
- It is a component that runs in the background to perform long-running operations or to perform work for remote processes
- A service does not have a user interface. For example, a service may play music in background while user is in a different app, or it might fetch data without blocking user interaction.
- Another component, such as an activity, can start the service and let it run or bind to it in order to interact with it.
- There are actually two very distinct semantic services tell the system about how to manage an app: Started services tell the system to keep them running until their work is completed. This could be to sync some data in background or play music even after user leaves the app.
- A regular background service is not something user is directly aware of, so system has more freedom in managing the process.
- Bound services run because some other app has said that it wants to make use of service.
- Live wallpapers, notification listeners, screen savers, etc. are all built as services that applications can implement & system binds to it when they should be running

Don Bosco Institute of Technology

Date:- 29/11/21 Branch:- IT Sem:- 7 Seat:- 1227933
Day:- Mon Sub:- MAD Sign:- Gupta
Time:- 3-5:30 Page:- 12/19

Q SQN

A service is implemented as a subclass of Service

```
public class MyService extends Service {
}
```

Receivers:-

- A receiver is a component that enables the system to deliver events to the app outside of regular flow, allowing the app to respond to system wide broadcast announcements.
- Because broadcast receivers are another well defined entry, the system can broadcast even to apps that are not running.
- for eg:- Alarm can be scheduled by an app to post a notification to tell the user about an upcoming event, there is no need for the app to keep running.
- Many broadcast originate from the system, eg:- broadcast announcing screen has turned off, battery is low, etc.
- Although, broadcast receivers don't have a user interface, they may create a status bar notification to alert the user when a broadcast event occurs.
- For instance, it might schedule a JobService to perform some work based on event with JobScheduler.

Don Bosco Institute of Technology, Kurla, Mumbai

Date:- 29/11/21 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:- Gupta
Time:- 3-5:30 Page:- 13/ 19

Q SOL

A broadcast receiver is implemented as a subclass of BroadcastReceiver and each broadcast is delivered as an Intent object

```
public class MyReceiver extends BroadcastReceiver {
    public void onReceive (context, intent) { }
}
```

4 B

(II) XML and JSON parsing in Android

- JSON is an independent data exchange format and is an alternative for XML
- Suppose the JSON data is given below & we are required to parse it and get temperature only.

```
{
    "sys" :
        { "country" : "GB"
          "sunrise" : 1234567,
          "sunset" : 1381149,
        },
    "weater" : [
        {
        "id" : 711
        "main" : "smoke"
        "description" : "smoke",
        "icon" : "50n"
        }
    ]
```

Don Bosco Institute of Technology, Kurla, Mumbai

Date:-29/11/21 Branch:-IT Sem:-7 Seat:-12277933
Day:-Mon Sub:-MAD Sign:-
Time:-3-5.30 Page:-M/19

Q | SON

```
"main" :
    {
        "temp" : 304.15
        "pressure" : 1009,
    }
}
```

JSON parsing: To parse a JSON object, create an object of class JSONobject & specify a string containing JSON data to it

```
String in,
JSONObject reader = new JSONObject(in)
JSONObject sys = reader.getJSONObject("sys");
country = sys.getString("country")

JSONObject main = reader.getJSONObject("main");
temperature = main.getString("temp");
```

XML Parser

XML is a popular parser format and used for sharing data on Internet. Android provides 3 types of XML parsers: DOM, SAX and XMLPullParser.

Consider data in XML format as follows:-

```
<?xml version = "1.0" ?>
<country>
    <city id="2643743" name="London">
    <coord lon = "-0.12574" Lat="51.0853" />
    <country> GB </country>
```

Don Bosco Institute of Technology, Mumbai

Date:- 29/11/24 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:-
Time:- 3-5.30 Page:- 15/19

of sen

```
</city>
<temperature value="285.54" min="284.15"
        max="295.15" unit=kelvin" />

<humidity value="71" unit="%" />
<pressure value="1015" unit="hPa" />

</country>
```

XML Parsing code

```
private XMLPullParserFactor XMLfactoryObject =
XMLPullParserFactory.newInstance();

private XMLPullParser myParser =
        XMLfactoryObject.newPullParser();
myParser.setInput(stream, null);

int event = myparser.getEventType();
while (event != XMLPullParser.END_DOCUMENT)
  {
    String name = myParser.getName()
    Switch (event){
        case XMLPullParser.START_TAG)
          break
        case XMLPullPar.END TAG,
          if (name.equals("temperature")) {

        temperature = myParser.getAttribute.
                    value(null, "value");
    }
```

Don Bosco Institute of Technology, Kurla

Date:- 29/4/24 Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Mon Sub:- MAD Sign:- Gupta
Time: 3-5.30 Page:- 16/19

Q SQN

```
            break;
        }
        event = myParser.next();
    }
```

4 A.

(1)

A service is a component that runs in background to perform long running operations without needing to interact with user.
It can take two states
(1) Started (2) Bound

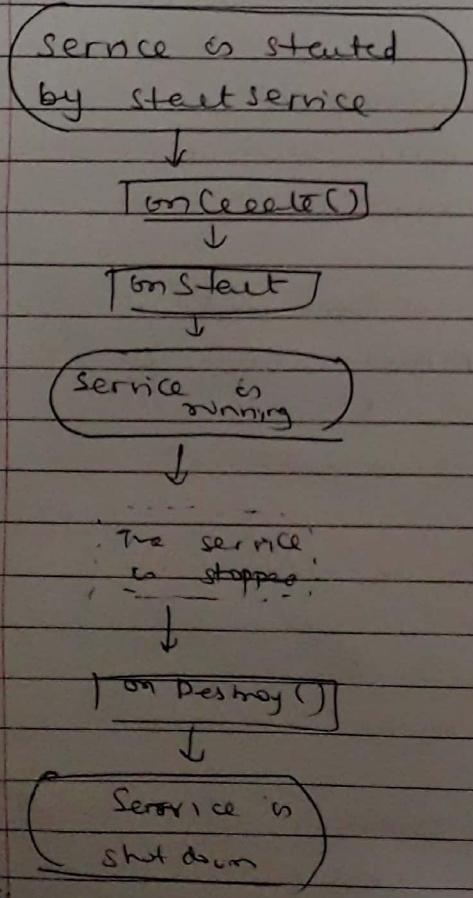

fig:- UN Bounded Service

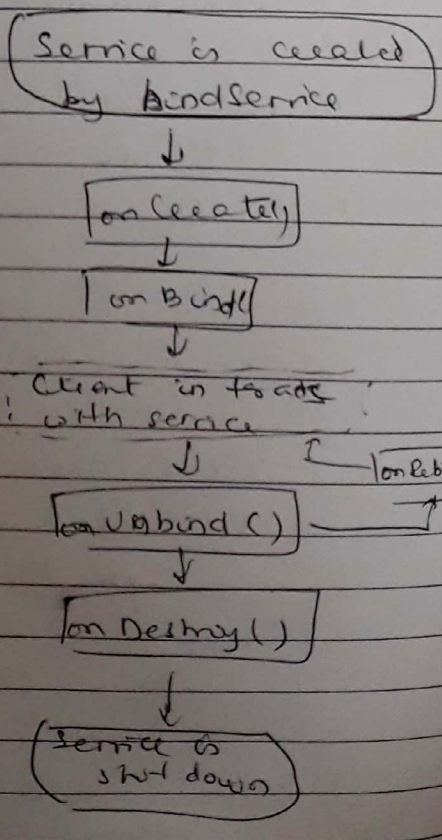

fig:- Bounded Service

Don Bosco Institute of Technology, Mumbai

Date:- Branch:- IT Sem:- 7 Seat:- 12277937
Day:- Sub:- MAD Sign:-
Time:- 3-5:30 Page:- 11 19

Q SOL

Various callback methods in Lifecycle of a service :-

(1) onStartCommand() :-
The system calls this method when another component requests that service be started.

(2) onBind() :-
The system calls this method when another component wants to bind with the service by calling bindService

(3) onUnBind() :-
The system calls this method when all clients have been disconnected from particular interface published by the service.

(4) onRebind() :-
The system calls this method when new clients have connected to the service, after it had been notified that all had disconnected on its onUnbind(Intent).

(5) onCreate() :-
The system calls this method when service is first created using onStartCommand() or onBind(). This call is required to perform one-time setup.

Date:- Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Sub:- MAD Sign:- Gupta
Time:- 3-5:30 Page:- 18/19

Q SQN

(6) onDestroy():-
The system calls this method when the service is no longer used and is being destroyed. The service should implement this to clean up any resources such as threads, registered listeners, receivers, etc.

Q4 A

(i) Broadcast receivers:-
A broadcast receiver is a component that enables the system to deliver events to the app, outside of regular user flow, allowing app to respond to system-wide broadcast announcements.

A Broadcast Receiver can be registered in two ways:-

By defining it in Androidmanifest.xml as shown below.

```
<receiver
    android:name=".ConnectionReceiver" >
    <intent-filter>
    <action
        android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    <intent-filter>
</receiver>
```

Don Bosco Institute of Technology, Mumbai

Date:- Branch:- IT Sem:- 7 Seat:- 12277933
Day:- Sub:- MAD Sign:- Rupta
Time:- 3-5.30 Page:- 19/19

Q SOL

Using intent fillers we tell the system that any intent that matches our ~~subfeats~~ subelements should get delivered to specific broadcast receiver.

(2) By defining it programmatically:-

```
IntentFilter filter = new IntentFilter();
intentFilter.addAction(getPackageName() +
        "android.net.conn.CONNECTIVITY_CHANGE");
MyReceiver myReceiver = new MyReceiver();
registerReceiver(myReceiver, filter);
```

Using the above code snippet, we can register broadcast receiver.