

Roll No: 19101A2006

SUB: MAD.

SEM: 7

Name: Harshala Adekar

12277642

1/19

Q.1

- ① Option D : Test /
- ② Option D - Dex Compiler
- ③ Option C - Loading an application using layout file
- ④ Option B - adb shell pm list permissions -S
- ⑤ Option A - Packaging and compiling Android Apps.
- ⑥ Option A - content://
- ⑦ Option C - Internal storage
- ⑧ Option C - Apktool Apktool
- ⑨ Option B - manifestxml Tool on Restart()
- ⑩ Option B - Starts an activity using an implicit intent.



FOR EDUCATIONAL USE

Harshala Adekar

Name : Jayshala Ardekar

219

Q.3

A. a. Java Virtual Machine & Dalvik Virtual Machine.

JVM (Java Virtual Machine)

① It is stack based VM that performs arithmetic & logic operations through push & pop operands. The result of operations is stored in stack memory.

② Java source code is compiled into java byte code format (.class file) that further translates into machine code.

③ In this, compiled bytecode size is compact because the location of the operand is simplicity on the operand stack.

DVM (Dalvik Virtual Machine)

① Register-based VM that uses registers located in the CPU to perform arithmetic & logic operations.

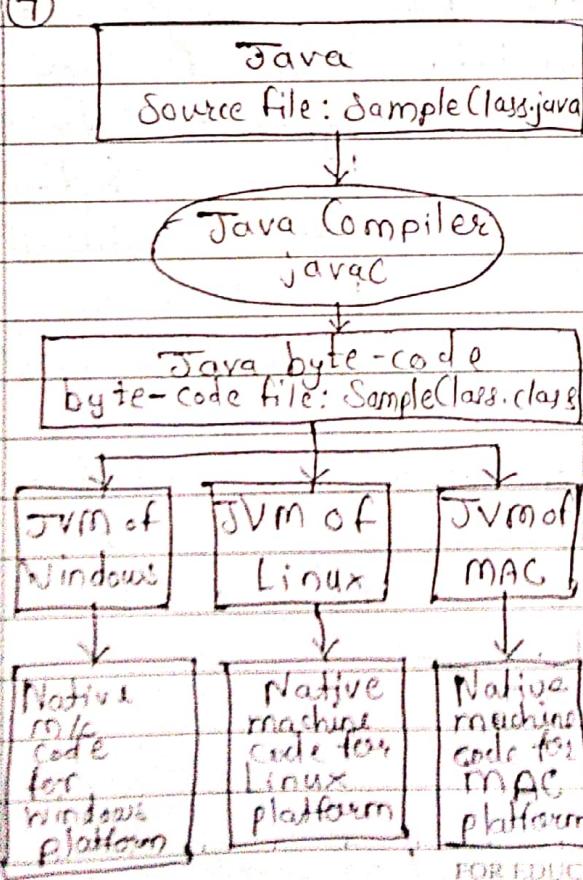
② Source code files are first of all compiled into java bytecode format like Jvri. Further, the DEX compiler (dx tool) converts the java bytecode into dalvik bytecode (classes.dex) file that will be used to create the .apk file.

③ Compiled bytecode size is larger as each instructions need all-implicit operands.

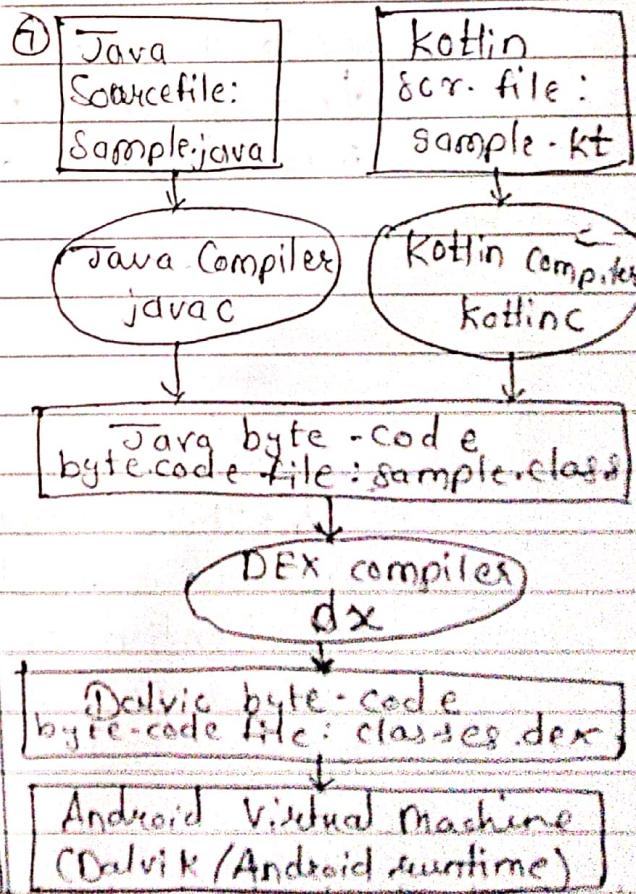
Q.S A)a.

- (1) The executable file for the device is .jar file.
- (2) The executable file for the device is .apk file.
- (3) A single instance of JVM is configured with shared process & memory space in order to run all deployed applications.
- (4) The device runs multiple VM instances with a separate process in shared memory space to deploy the code of each application.
- (5) Supports multiple operating systems like Linux, Windows & Mac.
- (6) Support only the Android operation system.

(7)



(8)



Q.3

A)b. Activity and Fragment

Activity	Fragment
① Activity is an application component that gives a user interface where user can interact.	① The fragment is only part of an activity, it basically contributes its UI to that activity.
② Activity is not dependent on fragment.	② Fragment is dependent on activity. It can't exist independently.
③ We need to mention all activity in the manifest xml file.	③ Fragment is not required to mention in the manifest file.
④ We can't create multi-screen UI without using fragments in an activity.	④ After using multiple fragments in a single activity, we can create a multi-screen UI.
⑤ Activity can exist without a fragment.	⑤ Fragment cannot be used without a activity.
⑥ Creating a project using only activity is difficult to manage.	⑥ While using fragments in the project, the project structure will be good and we can handle it easily.

Q.3

A)b.

⑦ Lifecycle methods are hosted by OS. The activity has its own lifecycle.

⑧ Activity is not light-weight.

⑦ Lifecycle methods in fragments are hosted by hosting the activity.

⑧ The fragment is light-weight.

B) → Implementing security in Android Apps :-

Making our app more secure helps us to preserve user trust & device integrity. Following are best practices for implementing security.

1) Enforce secure communication : When we safeguard the data that we exchange between our app and other apps, or between app and a website, we improve our apps' stability & protect the data that we send and receive.

2) Use of implicit intents & non-exported content providers :

→ If an implicit intent can launch at least two possible apps on a user's device, explicitly show in web browser or an app chooser.

→ This interaction strategy allows user to transfer sensitive information to an app.

Q.3 b) that they trust.

3) Apply signature-based permissions)

→ When sharing data between two apps that you control or own, use signature-based permissions.

→ These permissions don't require user confirmation and instead check that the apps accessing the data are signed using the same signing key.

→ Therefore, these permissions offer a more streamlined, secure user experience.

4) Disallow access to your app's content provider's:

→ Unless you intend to send your data from your app to a different app that you don't own, you should explicitly disallow other developer's apps from accessing the content provider objects that your app contains.

5) Ask for credentials before showing sensitive information:

→ When requesting credentials from users so that they can access sensitive information or premium content in your app, ask for either a PIN/password, such as their user's personal information.

Roll No : 19101A2006

SUB : MAD SEM : 7

Name : Hayhala Adekai,

12277642

7/19

Q.3 B) :-

⑥) Apply network security measures :

- Use SSL traffic - If app communicates with a web server that has a certificate issued by a well-known, trusted CA, the HTTPS request is very simple.
- Add a network security configuration: If app uses new or custom CAs, you can declare your network's security setting in a configuration file.

Roll No :- 19101A2006
SUB :- MAD SEM :- 7
Name :- Harshala Audekare,

12277642

8/12

Q.4 A)

① Broadcast receivers :

- ① Broadcast receiver simply respond to broadcast messages from other applications or from the system itself.
- ② These messages are sometime called events or intents.
- ③ For ex. applications can also initiate broadcasts to let another applications know that some data has been downloaded to the device and is available for them to use. So this is broadcast receiver that will intercept the communication and will initiate appropriate actions.

④ Registering broadcast receivers :

→ An application listens for specific broadcast intents by registering a broadcast receiver in Android Manifest.xml file.

Code :

```
<receiver android:name = "MyReceiver">
    <intent-filter>
        <action android:name = "android.intent.action.
            BOOT_COMPLETED">
            </action>
        </intent-filter>
    </receiver>
```

(9/19)

Q.4

A)

(i) → ⑤ Different ways of registering broadcast receivers:

a) android.intent.action.CALL

→ perform a call to someone.

b) android.intent.action.REBOOT

→ Have the device reboot.

c) android.intent.action.DATE_CHANGED

→ The date has changed.

d) android.intent.action.BUG_REPORT

→ Show activity for reporting bug.

(ii) → Callback methods in the lifecycle of Service Component:

① A service has a lifecycle callback methods that you can implement to monitor changes in the service's state & you can perform work at the appropriate stage.

② Following are the callback methods :

1) onStartCommand() :

→ The system calls this method when another component, such as an activity, requests that the service has been started by calling startService().

Roll No :- 19101A2006
SUB : MAD SEM :- 7
Name : Harshala Adekar

12277642

10/19

Q.3

A)

①) OnBind():

→ The system calls this method when another component wants to bind with the service by calling bindService().

②) OnUnbind():

→ The system calls this method when all clients have disconnected from a particular interface published by the service.

③) OnRebind():

→ The system calls this method when new clients have connected to the service, after it had disconnected in its onUnbind(intent).

④) onCreate():

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

⑤) onDestroy():

→ The system calls this method when the service is no longer used and is being destroyed. Your service should implement this to clean up any resources such as threads, registered listeners, receivers, etc.

Roll No : 19101A2006

SUB : MAD

SEM : 7

12277642

Name : - Harshala Ardekar

11/19

Q.4

5)

(ii) XML and JSON parsing in Android :

① JSON is an independent data exchange format and is an alternative for XML.

② Suppose the JSON parsing data is given below and we require to parse it to get temperature only :

{

"sys" :

{ "country" : "GB"

 "sunrise" : 1381107633

 "sunset" : 1381149604

}

"weather" : [

{ "id" : 711

 "main" : "smoke"

 "description" : "smoke"

 "icon" : "50n"

}

[

"main"

{ "temp" : 304.15,

 "pressure" : 1009,

}

[

③ JSON parsing : To parse a JSON object, create an object of class JSONObject and

Roll No :- 19101A2006

SUB : MAD

SEM : 7

12277642

Name : Harshala Adekar

Q.4

B)

(ii) → 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");

④ XML parser : XML is a popular format and used for sharing data on internet. Android provides 3 types of XML parser, DOM, SAX and XMLPullParser. [Efficient and easy to use].

Consider data in XML format as follows :-

<?xml version = "1.0"?>

<country>

<city id = "2643743" name = "London">

<coord lon = "-0.12574" lat = "51.50853"/>

<country>.GB </country>

<sunrise = "2013-10-08T06:13:56" set =
"2013-10-08T17:21:05" />

</city>

<temperature value = "289.54" min = "289.15"
max = "290.15" unit = "kelvin" />

<humidity value = "77" unit = "%"/>

Roll No :- 19101A2006
SUB: MAD SEM: 7

12277642

Name: Harshala Atedekar

13/19

Q.4 B)

(ii) $\langle \text{pressure} \text{ value} = "1025" \text{ unit} = "hPa" \rangle$
 $\langle / \text{country} \rangle$

⑤ XML Parsing code:

Provide XMLPullParserFactory 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 XMLPullParser.END_TAG;
                if (name.equals("temperature"))
                    {
                        temperature = myParser.getAttributeValue(null, "value");
                    }
                break;
        }
        event = myParser.next();
    }
```

Name : Harshala Adekar

14/19

Q.2

- A) ① Views : All the interaction of a user with the Android application is through the User Interface (UI), hence it is very important to understand the basics about User Interface of an android application. Here view refers to the android.view.View class which is super class for all the GUI components like TextView, ImageView, Button, etc.
- ② View is the basic building block of UI (User Interface) in android.
- ③ View extends object class and implements Drawable, Callback, KeyEvent.Callback and AccessibilityEventSource.
- ④ View can be considered as a rectangle on the screen that shows some type of content. It can be image, a piece of a text, a button or anything that an android application can display.
- ⑤ The rectangle here is actually invisible, but every view occupies a rectangle shape.

Use of ListView to select programming language from a list of languages:-

Step 1: Create a new project

- ① Click on file, then New \Rightarrow New Project
- ② Choose "Empty Activity" for the project template
- ③ Select language as Java.
- ④ Select minimum SDK version as per requirement

FOR EDUCATIONAL USE

Hadekar

12277642

Roll No :- 19101A2006

SUB: MAD

SEM :- 7

Name : Harshala Audekar

15/19

(Q.2)

A) Step 2 : Modify activity_main.xml file

Add a ListView in the activity_main.xml file.

Code :

```

<?xml version = "1.0" encoding = "utf-8"?>
<LinearLayout
    xmlns:tools = "MainActivity">
    <ListView
        android:id = "@+id/list"
        android:layout_width = "match_parent"
        android:layout_height = "match_parent"/>
</LinearLayout>

```

Step 3 : Modify MainActivity.java fileCode :

```

import androidx.appcompat.app.AppCompatActivity;

```

```

import android.os.Bundle;

```

```

import android.widget.AdapterView;

```

```

import android.widget.ArrayAdapter;

```

```

import android.widget.ListView;

```

```

public class MainActivity extends AppCompatActivity {

```

```

    ListView l;
```

Audekar

Roll No. :- 19101A2006
SUB : MAP SEM : 7
Name: Harshala Adekar

12277642

16/19

Q.2 A)

String languages [] = { "Java", "C", "C++", "DSA",
"Python" };

@Override

```
protected void onCreate(Bundle savedInstanceState)
{ super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    l = findViewById(R.id.list);
    ArrayAdapter<String> arr;
    arr = new ArrayAdapter<String>(this,
        R.layout.support_simple_spinner-
        dropdown_item, languages);
    l.setAdapter(arr);
```

}

}

Output :-

ListView
Java
C
C++
DSA
Python

FOR EDUCATIONAL USE

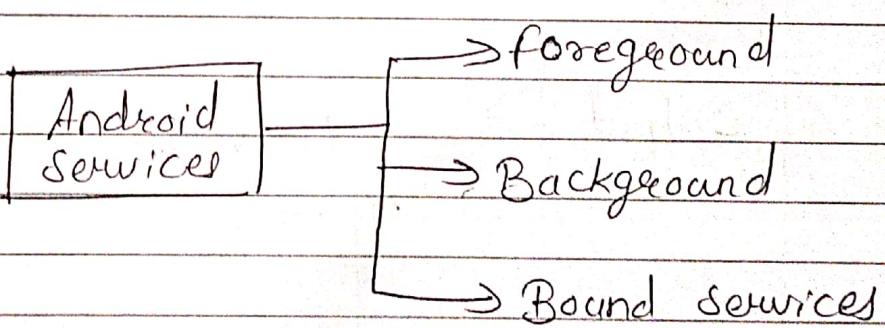
J. Adekar

(Q.2)

B) →

④ Service:

- ① Services in Android are special component that facilitates an application to run in the background in order to perform long-running operation tasks.
- ② The prime aim of service is to ensure that the application remains active in the background so that the user can operate multiple applications at the same time.
- ③ A service can run continuously in the background even if the application is closed or the user switches to another application.
- ④ There are three types of android services.



a) foreground services: They notify the user about its ongoing operations are termed as foreground service.

b) Background services: They do not require any user intervention. These services do not notify the user about ongoing background task & user also can not

18/19

Q2 B) access them

c) Bound Services : This service allows the components of the application like activity to bind themselves with it.

- Example of Android Service :

→ Playing music in background is a very common ex. of service in android.

code :

```
public class MyClass extends Service
{
    private MediaPlayer player;
    @Override
    public int onStartCommand(Intent i, int
        flags, int startid)
    {
        player = MediaPlayer.Create(this,
            setting.system.Default.Ring
            TONE_URI);
        player.start();
    }
}
```

- Broadcast Receivers :

① Broadcast in android is the system-wide events that can occur when device starts when the message is received on the device or when incoming calls are received or when a device goes to airplane mode etc.

19/19.

Q.2 (b) ② Broadcast receivers are used to respond to these system-wide events.

Ex. of Broadcast Receiver :

① Creating the broadcast receiver :

```
class MyClass extends BroadcastReceiver {  
    public void onReceive(Context c, Intent i)  
    {  
        if (i.getAction().equals("android.intent.action.AIRPLANE_MODE_CHANGED"))  
        {  
            Toast.makeText(c, "Airplane mode on",  
            Toast.LENGTH_LONG).show();  
        }  
        else  
        {  
            Toast.makeText(c, "Airplane mode off",  
            Toast.LENGTH_LONG).show();  
        }  
    }  
}
```

private static boolean isModeOn(Context c)
{
 return Settings.System.getInt(c.getContentResolver(),
 Settings.Global.AIRPLANE_MODE_ON) != 0;
}

② Registering Broadcast Receiver :

Code :- protected void onStart() {
 super.onStart();
}

```
IntentFilter f = new IntentFilter(  
    Intent.ACTION_AIRPLANE_MODE_CHANGED);  
registerReceiver(obj, f);
```

(obj is the class created above.)