

Assignment :- 2

Date _____
Page _____

Q: 1 what is event driven programming? what are the benefits of that?

- With the development of program with GUI normally you want something to happen when the user performs certain actions, for Example:
 - clicking on Button.
 - Entering text into a TextField.
- For a program to respond to events, there are a no. of parties involved in events and event handling.
- So, Any application or program, in which we have to interact with UI like touch, press, click, then it is known as event driven programming.

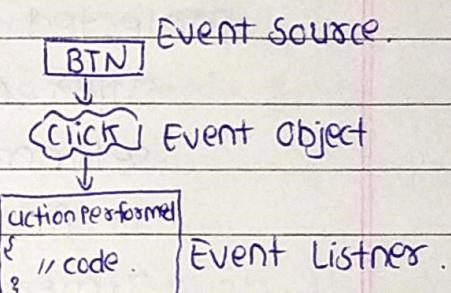
→ These are mainly 3 parties.

(1) event object

(2) event source

(3) event listener.

Ex:-



- Android is an event driven programming

→ Benefits of it :-

- Flexibility
- Suitable for Graphical Interfaces.
- Simplicity of Programming
- Easy to find Natural Dividing lines.
- Highly Composable.
- Simple & Understandable.
- Purely Procedural.
- Using Hardware Interactive.
- Allows Sensors.

Q:2

Write android app code for change app colour after every five seconds.

→

```
package com.example.bgchange;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.graphics.Color;
```

```
import android.os.Bundle;
```

```
import java.util.Random;
```

```
import java.util.Timer;
```

```
import java.util.TimerTask;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    LinearLayout l;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    l = findViewById(R.id.lin);
```

```
    Timer timer = new Timer();
```

```
    MyTimer myTimer = new MyTimer();
```

```
    timer.schedule(myTimer, 5000, 2000);
```

```
}
```

```
public class MyTimer extends TimerTask {
```

```
    public void run() {
```

```
        runOnUiThread(new Runnable() {
```

```
            @Override
```

```
            public void run() {
```

```
                Random r = new Random();
```

```
                l.setBackgroundColor(Color.rgb(255,
```

```
                r.nextInt(256), r.nextInt(256)));
```

```
}
```

```
} } );
```

```
3
```

Q:3 What is android debug? Explain with example?

- An Android Debug Bridge is a command line tools which can be used to communicate with Emulator or Android Devices.
- When your development machine needs to communicate with an Android Device it does so using ADB. It's a process that is controlled by a command also known as ADB.
- The ADB command works by talking to an and server which runs under in background at port 5037. Android Studio also talks to this server when it needs to run an app via an Android Device.
- To work with ADB you need Android SDK.

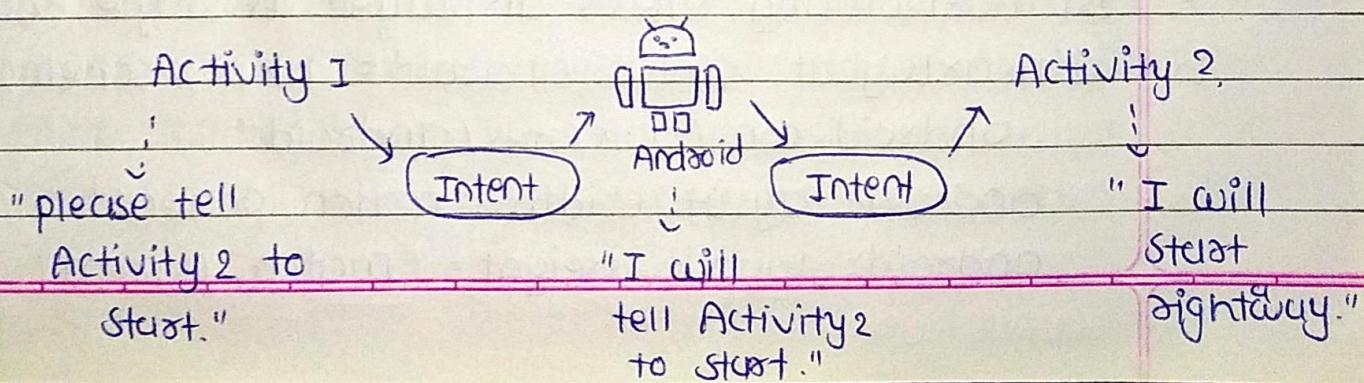
Ex: For debug an application, we use Android Emulator (Virtual Device) or Android Device.

- For Debug app in Android Device.
- First we have to connect Device through USB cable & turn on USB DEBUGGING
- After that you can easily run your application.

Q:4 What is intent? Explain with examples?

- An intent is a type of message.
- The intent specifies the activity you want to receive it.
- It's like putting an address on an envelope.

Ex:-



Ex - Intent intent = new Intent(this, Activity2.class);
 startActivity(intent);

→ Applications of Intent :

- Sending the user to another app.
- Getting a Result from an Activity.
- Allowing other apps to start your Activity.

Ex - For use an action- Activity.

- in java file.
- when we clicked on SEND button, it shows suggestion apps.

Ex - public void onBtnClick(View view) {

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.setType("text/plain");
intent.putExtra(Intent.EXTRA_TEXT, msg);
startActivity(intent);
```

}

Q:5 Explain Linear Layout and Relative Layout with Ex.?

→ Linear Layout :

- It displays views next to each other either vertically or horizontally.
- In vertically, views displayed in single column.
- In Horizontally, views displayed in single row.

Ex - <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

 android:layout_width="match_parent"

 android:layout_height="match_parent"

`android: orientation = "vertical" ...>`

`</LinearLayout>`

→ Relative Layout:

- It displays its views in relative position.
- You define the position of each view relative to other views in the layout, or relative to its parent layout.
- As an example you choose to position a text view relative to the top of the parent layout, a spinner under the text view and a button relative to bottom of the parent layout.

Ex: `<RelativeLayout ...>`

`<TextView`

`android: id = "@+id / txt" />`

`<Button`

`...`

`android: layout_below = "@+id / txt" ... />`

`</RelativeLayout>`

Q:6 What is splash screen? Write suitable code!

- It enables a new app launch animation for all apps when running on a device with Android 12 or Higher.
- This includes an intro-app motion at launch, a splash screen showing your app icon, and a transition to your app itself.
- First screen of the app when it is opened.

Ex :-

We want to add splash screen in our app.

For that we have to design an Empty screen for splash screen and we have to remove action bars from them.

- `supportActionBar?.hide() // For hide action bars.`
- Now in splash activity, we have to set intent & handles in .java file.

In `onCreate()`:

`Handler().postDelayed({`

`val intent = Intent(this@MainActivity, Home
Activity::class.java)`

`startActivity(intent)`

`}, 3000);`

- Now, our splash screen works.

Q:7 Why does standard java byte code not run in android app? (Give Example).

→ Following Reasons:

- Android uses Dalvik VM instead of JVM.
- To run Java Bytecode you need JVM.
- JAVA in computers and Android uses a separate environment to run their code. Android has been modified to run on smaller devices with the exhaustion of less computing power.
- In Android we have to convert Java class into Dalvik executable files using an android tool called dac.

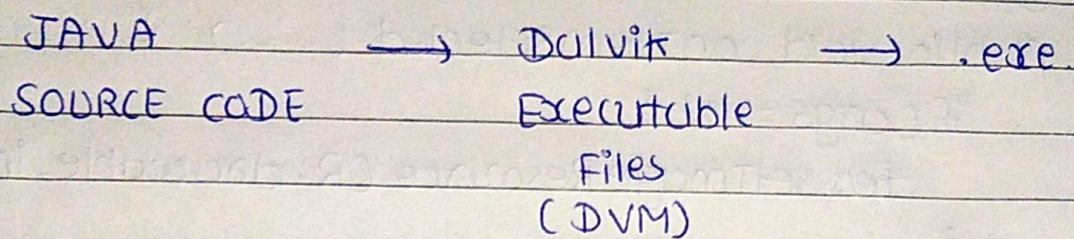
Ex:- In simple JAVA :

JAVA

SOURCE CODE

→ Byte Code → JVM → .exe

- JVM is required to convert byte code.
- But in Android studio:



Q:8 Write a code for a spinner app?

- Here we make one simple app using spinner in which by changing img spinner value, image will change.
- .xml File :

```
<RelativeLayout ...>
    <ImageView ... />
    <Spinner
        android:id="@+id/spinner"
        android:entries="@array/img" />
</RelativeLayout>
```

- .java File :

```
import android.*;
```

```
public class MainActivity extends AppCompatActivity
    implements AdapterView.OnItemSelectedListener {
    Spinner sp;
    ImageView iv;
    @Override
    protected void onCreate(...) {
        ...
        sp = (Spinner) findViewById(R.id.spinner);
        sp.setOnItemSelectedListener(this);
```

```
iv = findViewById (R.id.imgv);
```

```
}
```

```
@Override
```

```
public void onItemSelected (...) {
```

```
if (cargs == 0) {
```

```
iv.setImageResource (R.drawable.img);
```

```
}
```

```
if (cargs == 1) {
```

```
iv.setImageResource (R.drawable.img_1);
```

```
}
```

```
}
```

→ string.xml :

```
<string-drawable name="img">
```

```
  <item> J1 </item>
```

```
  <item> J2 </item>
```

```
</string-drawable>
```

Q: 9 What is animation in android app? Give suitable example with code.

→ Animation is the process of adding a motion effect to any view, image or text. With the help of an animation, you can add motion or can change the shape of specific view.

- Animation in Android is generally used to give your UI a rich look and feel.

- There are basically 3 types.

- (1) Property Animation (3) Drawable Animation
- (2) View Animation

Ex:- Here we make one animation which translate the image. For that we have to make one animation folder in res and create one animation file in it.

→ animation.xml File :- or translate.xml File :-

<set ...>

```
<translate android:fromXDelta="0"
           android:toXDelta="100"
           android:fromYDelta="0"
           android:toYDelta="100"
           android:duration="5000" /> </set>
```

→ activity_main.xml :-

<RelativeLayout ...>

<Spinner ... />

<ImageView ... />

</RelativeLayout>

→ java File :-

- Here we write code on "OnItemSelected"

public void OnItemSelected (....) {

if (arg3 == 1) { }

if (arg3 == 2) { }

Animation a = AnimationUtils.loadAnimation (this,

img.startAnimation (a);

R.a.translate);

}

}

→ String.xml File:-

<string - class name = "animation">

<item> None </item>

<item> Translate </item>

</string - class>

* Now Image can translate & change its position. (Image will Animate)