

### Q1. Name and explain the seven callback methods defined by the activity class?

### Q2. List and explain activity methods?

1. **OnCreate()**: يتم إنشاؤها عند استدعاء النشاط لأول مرة
2. **OnStart()**: يتم استدعاءها لتعيين قيمة البيانات أو لإعادة تعيين قيمة بيانات النشاط
3. **OnResume()**: يتم استدعاءها عندما يأتي نشاط في مقدمة نشاط مرئي آخر، حيث يظهر للمستخدم ويمكن التفاعل معه
4. **OnStop()**: يتم استدعاءها عندما يأتي نشاط غير مرئي للمستخدم، ولن يكون النشاط ظاهر بعد الآن
5. **OnPause()**: يتم استدعاءها عندما يأتي نشاط آخر في قمة الأنشطة
6. **OnRestart()**: يتم استدعاءها بعد أمر **OnCreate()**
7. **OnDestroy()**: يتم استدعاءها لتلقي آخر مكالمات مع النشاط، أي في نهاية دورة حياته

### Q3. Explain and draw the Android build system (Gradle)? \*Gradle Draw p16 chapter1

هي مجموعة من الأدوات Toolkit التي يتم استخدامها لـ run, test, build, package التطبيق، يمكن تشغيلها من قائمة Android studio أو بشكل مستقل من الcommand line، يمكن استخدام مميزات مثل:

1. تخصيص وتكوين وتوسيع عمليات البناء
  2. إعادة استخدام التعليمات
  3. إنشاء ملفات apk متعددة
1. **aapt**: Android Asset Package Tool, R.java ويترجمه الى (mainfest xml, java)
  2. **Aidl**: android interface definition language , (Java interface) يأخذ الواجهة ويحولها الى
  3. **Jarsigner**: يتم التوقيع على ملفات apk. قبل التثبيت على الجهاز
  4. **Zipalign**: هي أداة توفر تحسينات لملف apk تؤدي الى الهدف النهائي مع الحد من استخدام الذاكرة أثناء تشغيل التطبيق

### Q4. What is Intent, explain the types of Intents?

هي الجسر الذي يربط بين الActivities ، يمكن ان تخزن parameters وتمررها من نشاط الى نشاط ويمكن ان تسترجع قيمة لها هيكلية (Data, Action) ، تستخدم لبث الرسائل وبدء الخدمة. وهي اساسية في الاندرويد للربط وتمرير المعلومات لها نوعان:

1. **Implicit**: لا يتم اعطاء اسم ال class بل يتم الاعلان عن إجراء عام يتم تنفيذه ومعالجته بواسطة تطبيق اخر  
مثل: يتم اعطاء ال user رابط للانتقال الى تطبيق Google:

```
Intent i = new Intent (Intent.ACTION_VIEW, Uri.parse("http://www.Google.com"));
StartActivity(i);
```

2. **Explicit**: يتم الاعلان عن اسم ال class المراد الانتقال إليه عبر ال java class  
مثل: الانتقال من نشاط الى اخر عبر ال Button

```
Intent i = new Intent (this, Activity_name.class);
StartActivity(i);
```

### Q5. What is Broadcast Receiver? Draw p27 chapter1

هي مكونات في التطبيق تستمع الى البث، ولا تحتوي على واجهة استخدام وتقوم بإجراء معين مثل تغيير حالة الشبكة، استلام رسالة، التقاط صورة من كاميرة ..

### Q6. What are the forms of services?

1. **Started**: تبدأ عندما يستدعي مكون ما مثل النشاط (**StartService()**) وتنفذ عملية واحدة فقط، ولا ترد نتيجة الى المكون
2. **Bound**: تبدأ عندما يتم استدعاءها بواسطة مكون (**bindservice()**)، توفر clint-server interface تسمح للمستخدم بالتفاعل مع الخدمة وارسال طلب واستقبال نتيجة وتسمح للاتصال بين العمليات بواسطة IPC، لاتعمل الا بارتباط مكون آخر بها

### Q7. What is the purpose of R.java file? Briefly explain?

يتم انشاء هذا الملف عن طريق دالة aapt التي تأخذ ال Application Resource وتحواله الى R.java الذي يسهل عليك الرجوع الى مصدر كود java الخاص بالتطبيق ولكي يتم تحوله الى class. عن طريق Java compiler

## Q8. List and explain Android application component ?

### 1. Activity:

يمثل النشاط بواجهة واحدة (java+xml)، تعمل الأنشطة معاً لتشكيل تجربة المستخدم، وتكون الأنشطة مستقلة ويمكن البدء من أي نشاط على حسب التطبيق، النشاط الرئيسي main يبدأ في وقت lunchtime، يتكون البرنامج من مجموعة من الأنشطة

### 2. Services:

هو مكون يعمل على الخلفية يستخدم في العمليات طويلة المدى أو البعيدة ولا توفر واجهة مستخدم مثل: تشغيل برنامج موسيقى في الخلفية بينما تعمل على برنامج آخر

### 3. Intent:

هي الجسر الذي يربط بين الـ Activitys ، يمكن أن تخزن parameters وتمررها من نشاط إلى نشاط ويمكن أن تسترجع قيمة لها هيكلية (Data, Action) ، تستخدم لبث الرسائل وبدء الخدمة. وهي أساسية في الأندرويد للربط وتمرير المعلومات

### 4. Broadcast Receiver:

هي مكونات في التطبيق تستمع إلى البث، ولا تحتوي على واجهة استخدام وتقوم بإجراء معين مثل تغيير حالة الشبكة

### 5. Content provider: Draw p27

هو مكون يدير مجموعة مشتركة من البيانات ويقوم بتخزينها في ملفات ثابتة DB->SQLite أو على أي موقع web يوفر إمكانية القراءة و الكتابة

## Q9. Explain and draw the three different lifecycle of activity? P2 C2

1. Entire lifetime cycle: OnCreate() تبدأ بـ OnDestroy() تنتهي بـ
2. Visible lifetime cycle: OnStart() تبدأ بـ OnStop() تنتهي بـ
3. Foreground lifetime cycle: onPause() تبدأ بـ onResume() تنتهي بـ

## Q10. illustrate how an implicit intent is delivered through the system to start another activity? C3 P7

1. يقوم الـ Activity 1 بإنشاء Intent ووصف الإجراءات وتمريرها إلى startActivity()
2. يقوم الـ Android system بالبحث في جميع التطبيقات عن Intent مطابق من خلال Intent filter
3. عند إيجاد التوافق يقوم النظام ببدء Activity 2 بإستدعاء onCreate() وتمريرها إلى Intent

## Q11. Explain the Android project structure?

1. Android Manifest-xml: يحتوي على التكوين العام للمشروع والمعلومات الأساسية التي يحتاجها الأندرويد قبل تشغيل التطبيق
2. Src/java: كود المصدر الخاص بالجافا
3. Res: resource files (many are XML)
  - Drawable/ image
  - Layout/ description of GUI layout
  - String/ بيانات الترجمة
  - Style/ الشكل العام للتصميم
4. Gradle: build.Gradle -> compile management system , Gradle.build -> main build config file

## Q12. Explain intents type, give example of how to create intent and send it to the Android system?

```
Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.Google.com"));
startActivity(i);
```

## Q13. Draw and list and explain the Activity states? P5 C2

1. Start state: SetUp عمل الـ Activity
2. Running/Resumed state: ظاهرة وتتفاعل مع المستخدم، يعتبر النشاط في حالة running إذا كان في المقدمة
3. Paused state: ظاهرة ولا تتفاعل مع المستخدم، يمكن إيقافها بشكل جزئي أو كلي، ويمكن عمل لها running من جديد
4. Stopped state: غير ظاهرة ويمكن إيقافها وهي نشاط متوقف في الخلفية بأولوية أقل

#### Q14. Draw life cycle? P6 C2

#### Q15. Explain Activity Stack? P8 C2

- عندما يأتي نشاط فوق آخر، يكون النشاط الجديد في أعلى الـ Stack
- يبقى النشاط السابق في أسفل الـ Stack لحين انتهاء النشاط الحالي
- يتم تفعيل النقل الى الامام/الخلف بإجراء من الـ user

#### Q16. What is Android?

1. Mobile OS maintained by Google
2. The 1 mobile OS worldwide
3. Based on java and Linux

#### Q17. Explain what is Event? and explain event-driven programming?

هو عامل خارجي يمكن للتطبيق الاستجابة له عن طريق كتابة دالة Listener وله انواع شائعة:

- حركة الماوس ، الضغط على المفاتيح ، انتهاء صلاحية المؤقت

يتم التحكم في تنفيذ البرنامج من خلال الاحداث التي تحدث، ويشيع استخدامه في البرامج الرسومية: **event-driven programming**

#### Q18. Explain what is Toasts?

هو رسالة تظهر لفترة قصيرة على شكل pop-up ، مفيدة لعرض التحديثات استجابةً للإحداث التي فعلها المستخدم

```
Toast.mackText(this, "message", duration).show();
```

Duration is Toast.LENGTH\_SHORT or LENGTH\_LONG

#### Q19. How to pass parameters?

```
Intent i = new Intent (this, Activity_name.class);
```

```
i.putExtra ("FName", value);
```

```
i.putExtra ("LName", value);
```

```
StartActivity(i);
```

#### Q20. How to Extract data?

```
Intent i = getIntent();
```

```
String FirstName = i.getStringExtra("name1");
```

```
String LastName = i.getStringExtra("name2");
```

#### Q21. Sending back Result?

```
public void onClick(View view) {
```

```
Intent i = new Intent();
```

```
String message = "abc";
```

```
intent.putExtra("MESSAGE",message);
```

```
setResult(2,i);
```

```
finish();
```

```
} }
```

#### Q22. The Android platform supports Background processing in 4 different ways?

1. Thread: يدعم استخدام thread تنفيذ الاجراء بشكل غير متزامن
2. Handler: يمكن من خلاله handler تحديث واجهة المستخدم ويوفر طرق تلقي الرسائل
3. Async Task: فئة خاصة بالاندرويد تغلف معالجة الخلفية و تسهل اتصال وتحدث واجهة المستخدم
4. Service: هي مكون يعمل في الخلفية لاداء عمليات طويلة الأمد دون حاجة لتفاعل المستخدم

#### Q23. How to create Thread?

1. Implement Runnable interface:

```
Runnable myRunnable = new MyClassRunnable();
Thread t = new Thread (myRunnable);
t.Start();
```

2. Extends Thread Class:

```
MyThread t = new MyThread();
t.start();
```

#### Q24. MultiThread programming challenges include?

1. Overriding data
2. Deadlock
3. اختبار وتصحيح
4. اعتماد على البيانات
5. تقسيم عبء العمل

#### Q25. Thread Class Methods?

1. معرفة اسم الخيط: getName()
2. الحصول على الاولوية: GetPriority()
3. معرفة الخيط هل هو على قيد التشغيل: IsAlive()
4. انتظر حتى ينتهي الخيط: Join()
5. نقطة دخول للخيط: Run()
6. ايقاف مؤقت للخيط: Sleep()

#### Q26. Why using Multi-Threads?

1. اذا كان وقت تنفيذ مؤشر الترابط اعلى من 5 ثوان فسيقوم OS بعرض رسالة خطأ ANR
2. المهام البطيئة مثل تنزيل الملفات لا يمكن تشغيلها على ال-main-Thread انما تستخدم Multi-Thread
3. Multi-core-cpu تعمل على جعل Thread يعمل بالتوازي

#### Q27. How to use Multi-Thread?

1. Classical Thread Programming
2. Main thread only update the UI

#### Q28. What is Broadcast Receivers, give an example on how to set up a Broadcast Receivers in android app?

هو مكون يسمح لك بالتسجيل في احداث النظام او التطبيق ، ال receiver هو رمز غير نشط يتم تنشيطه بمجرد حدوث حدث مشترك فيه، وهو abstract class وعنده onRecevier() method لكي يتم اعداد ال Broadcast receiver نحتاج الى:

1. creating BroadcastReceiver
2. Registering BroadcastReceiver

#### وللبداء في التسجيل Registering BroadcastReceiver:

1. By defining it in the AndroidManifest.xml file.
2. By defining it programmatically.

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

#### Q29. Service Life Cycle Methods?

1. OnUnbind(): يستدعي النظام هذه العملية عندما يفصل جميع العملاء عن واجهة معينة للخدمة
2. OnRebind(): يستدعي النظام هذه العملية عندما يتصل عملاء جدد بالخدمة، بعد انفصال العملاء السابقين
3. OnCreate(): يستدعي النظام هذه العملية عند انشاء الخدمة لأول مره
4. OnDestroy(): يستدعي النظام هذه العملية لتدمير الخدمة

#### Q30. Service - Life Cycle:

1. **The entire lifetime of a service:** happens between the time `onCreate()` is called and the time `OnDestroy()`.
2. **The active lifetime of a service:** begins with a call to either `onStartCommand()` ends with `OnDestroy()` or `onBind()` ends when `onUnbind()`.

### Q31. Explain how AsyncTask defined?

By 3 generic types : Params, Progress, Result

And 4 steps:

1. `onPreExecute`
2. `doInBackground`
3. `onProgressUpdate`
4. `onPostExecute`

### Q32. What is AsyncTask, explain?

تتيح الاستخدام السليم للـ UI Thread تسمح هذه الفئة بإجراء عمليات في الخلفية ونشر النتائج على UI Thread دون الحاجة الى معالجة `thread/handlers`  
تم تصميمه لكي يكون فئة مساعدة للـ `thread/handlers`  
يجب استخدامه للعمليات القصيرة، اذا اردت بقاء الـ `thread` مشغول لوقت طويل

### Q33. Explain the steps AsyncTask Goes through when it is executed?

1. **onPreExecute:** يتم استدعاؤه قبل تنفيذ المهمة بشكل مثالي قبل استدعاء طريقة `doInBackground` على مؤشر ترابط واجهة المستخدم. تستخدم هذه الطريقة عادة لإعداد المهمة مثل إظهار شريط التقدم في واجهة المستخدم.
2. **doInBackground:** يجب وضع الرمز الذي يعمل لفترة طويلة الأمد في طريقة `doInBackground`. عند استدعاء طريقة التنفيذ في مؤشر الترابط الرئيسي لواجهة المستخدم، يتم استدعاء هذه الطريقة مع الملمات التي تم تمريرها.
3. **onProgressUpdate:** يتم استدعاؤها عن طريق `publishProgress` في اي وقت من `doInBackground` يمكن استخدام هذه الطريقة لعرض اي شكل من اشكال التقدم في واجهة المستخدم.
4. **onPostExecute:** يتم استدعاؤها بعد انتهاء العملية في الخلفية للـ `doInBackground` ويتم ارسال النتيجة للـ `doInBackground`.

### Q34. When to use AsyncTask? Draw C7 P4

### Q35. How is as AsyncTask, Thread, Service are similar? And how they are different?

### Q36. AsyncTask Rules?

1. class must be loaded on the UI thread.
2. task instance must be created on the UI thread.
3. Method execute must be invoked on the UI thread.
4. Should not call Methods manually.
5. The task can be executed only once

### Q37. When an Asynchronous Task is executed, the task goes through 4 steps? Draw in C7 P8

1. **onPreExecute:** تستخدم لاعداد العملية او الإجراء
2. **doInBackground(params):** يتم استدعاء هذه العملية عندما يتم تمرير البارامترات
3. **onProgressUpdate(progress):** يتم استدعاء هذه العملية لعرض اشكال التقدم وتطور العملية في واجهة المستخدم
4. **onPostExecute(Result):** يتم تمرير نتائج الـ `doInBackground` الى هذه العملية لاكمال المعالجة

### Q38. main uses for a Handler?

1. schedule messages and runnable to be executed as some point in the future.
2. enqueue an action to be performed on another thread

### Q40. Thread Interacting with the UI? Draw C6 P4

1. يقوم UI thread بإنشاء Handler object داخلي لنفسه
2. يستخدم main thread الـ object للحصول على رسالة فارغة وارسال رسالة الى UI thread