参考答案：

安卓篇

一、选择题

1.C 2. AC 3. AB 4. AD 5. C

二、问答题

1. Android系统采用了分层架构，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和linux核心层。

2. 在Android中，共有五种布局方式，分别是：FrameLayout(框架布局)，LinearLayout (线性布局)，AbsoluteLayout(绝对布局)，RelativeLayout(相对布局)，TableLayout(表格布局)。

（1）FrameLayout 框架布局,放入其中的所有元素都被放置在最左上的区域，而且无法为这些元素指定一个确切的位置,下一个子元素会重叠覆盖上一个子元素，适合浏览单张图片。

（2）LinearLayout 线性布局,是应用程序中最常用的布局方式，主要提供控件水平或者垂直排列的模型，每个子组件都是以垂直或水平的方式来定位.(默认是垂直)

（3）AbsoluteLayout 绝对定位布局,采用坐标轴的方式定位组件，左上角是（0，0）点，往右x轴递增，往下Y轴递增,组件定位属性为android:layout\_x 和 android:layout\_y来确定坐标。

（4）RelativeLayout 相对布局,根据另外一个组件或是顶层父组件来确定下一个组件的位置。和CSS里面的类似。

（5）TableLayout 表格布局,类似Html里的Table.使用TableRow来布局，其中TableRow代表一行，TableRow的每一个视图组件代表一个单元格。

3. Android提供了5种方式存储数据：

（1）使用SharedPreferences存储数据；它是Android提供的用来存储一些简单配置信息的一种机制，采用了XML格式将数据存储到设备中。只能在同一个包内使用，不能在不同的包之间使用。

（2）文件存储数据；文件存储方式是一种较常用的方法，在Android中读取/写入文件的方法，与Java中实现I/O的程序是完全一样的，提供了openFileInput()和openFileOutput()方法来读取设备上的文件。

（3）SQLite数据库存储数据；SQLite是Android所带的一个标准的数据库，它支持SQL语句，它是一个轻量级的嵌入式数据库。

（4）使用ContentProvider存储数据；主要用于应用程序之间进行数据交换，从而能够让其他的应用保存或读取此Content Provider的各种数据类型。

（5）网络存储数据；通过网络上提供给我们的存储空间来上传(存储)和下载(获取)我们存储在网络空间中的数据信息。

4. 一个activity呈现了一个用户可以操作的可视化用户界面。

一个service不包含可见的用户界面，而是在后台无限地运行。

可以连接到一个正在运行的服务中，连接后，可以通过服务中暴露出来的借口与其进行通信

一个broadcast receiver是一个接收广播消息并作出回应的component，broadcastreceiver没有界面。

intent:content provider在接收到ContentResolver的请求时被激活。

activity, service和broadcast receiver是被称为intents的异步消息激活的。

一个intent是一个Intent对象，它保存了消息的内容。对于activity和service来说，它指定了请求的操作名称和待操作数据的URI。Intent对象可以显式的指定一个目标component。如果这样的话，android会找到这个component(基于manifest文件中的声明)并激活它。但如果一个目标不是显式指定的，android必须找到响应intent的最佳component。它是通过将Intent对象和目标的intent filter相比较来完成这一工作的。

一个component的intent filter告诉android该component能处理的intent。intent filter也是在manifest文件中声明的。

5. Adapter是连接后端数据和前端显示的适配器接口。常见的Adapter有ArrayAdapter, BaseAdapter, CursorAdapter, HeaderViewListAdapter, ListAdapter, ResourceCursorAdapter, SimpleAdapter, SimpleCursorAdapter, SpinnerAdapter, WrapperListAdapter等。

苹果篇

1. 选择题

1.BCD 2.B 3.B 4.BCD 5.ABC

1. 问答题

1. assign用于简单数据类型，如NSInteger,double,bool。retain 和copy用于对象，copy用于当 a指向一个对象，b也想指向同样的对象的时候，如果用assign，a如果释放，再调用b会crash,如果用copy 的方式，a和b各自有自己的内存，就可以解决这个问题。retain 会使计数器加一，也可以解决assign的问题。另外：atomic和nonatomic用来决定编译器生成的getter和setter是否为原子操作。在多线程环境下，原子操作是必要的，否则有可能引起错误的结果。加了atomic，setter函数会变成下面这样：

if (property != newValue) { [property release]; property = [newValue retain]; }

2. 当您向一个对象发送一个autorelease消息时，Cocoa就会将该对 象的一个引用放入到最新的自动释放池。它仍然是个正当的对象，因此自动释放池定义的作用域内的其它对象可以向它发送消息。当程序执行到作用域结束的位置 时，自动释放池就会被释放，池中的所有对象也就被释放。

ojb-c 是通过一种”referring counting”(引用计数)的方式来管理内存的, 对象在开始分配内存(alloc)的时候引用计数为一,以后每当碰到有copy,retain的时候引用计数都会加一, 每当碰到release和autorelease的时候引用计数就会减一,如果此对象的计数变为了0, 就会被系统销毁。NSAutoreleasePool 就是用来做引用计数的管理工作的,这个东西一般不用你管的。 autorelease和release没什么区别,只是引用计数减一的时机不同而已,autorelease会在对象的使用真正结束的时候才做引用计数减一。

3. ios数据持久化方案包括：plist文件、preference、NSKeyedArchiver（归档）、SQLite3以及CoreData。

4. viewDidLoad在view 从nib文件初始化时调用，loadView在controller的view为nil时调用。此方法在编程实现view时调用,view 控制器默认会注册memory warning notification,当view controller的任何view 没有用的时候，viewDidUnload会被调用，在这里实现将retain 的view release,如果是retain的IBOutlet view 属性则不要在这里release,IBOutlet会负责release 。

5. 复用队列的元素增加只有在cell被滑动出界面的时候，此cell才会被加入到复用队列中。每次在创建cell的时候，程序会首先通过调用dequeueReusableCellWithIdentifier:cellType方法，到复用队列中去寻找标示符为”cellType”的cell，如果找不到，返回nil，然后程序去通过调用[[[UITableViewCell alloc] initWithStyle:style reuseIdentifier:cellType] autorelease]来创建标示符为”cellType”的cell。