

1. Explain the order of lifecycle changes in activity a and activity b as activity a starts activity b.

- Start the app
 - activity A: onCreate
 - activity A: onStart
 - activity A: onResume
- Push the button to go to Activity B
 - activity A: onPause
 - activity B: onCreate
 - activity B: onStart
 - activity B: onResume
 - activity A: onStop
- Push the button to go back to Activity A
 - activity B: onPause
 - activity A: onDestroy
 - activity A: onCreate
 - activity A: onStart
 - activity A: onResume
 - activity B: onStop
 - activity B: onDestroy
- Push the button to go home in the phone
 - activity A: onPause
 - activity A: onStop
- Cleaning memory in the phone
 - activity A: onDestroy

2. How does serialization work?

1. It translates the object's internal state into a stream of bytes
2. This binary stream or image of the object is created in an operating system-neutral network byte order
3. The image can be written to a disk, stored in memory, or sent over a network to a different operating system
4. This amazing feat requires little or no work on the part of the programmer
5. Just implement the serializable interface, which contains no methods, and call the writeObject() method on your object, and it's serialized! You can serialize an object to or from any I/O device that Java supports.

3. How does parcels work?

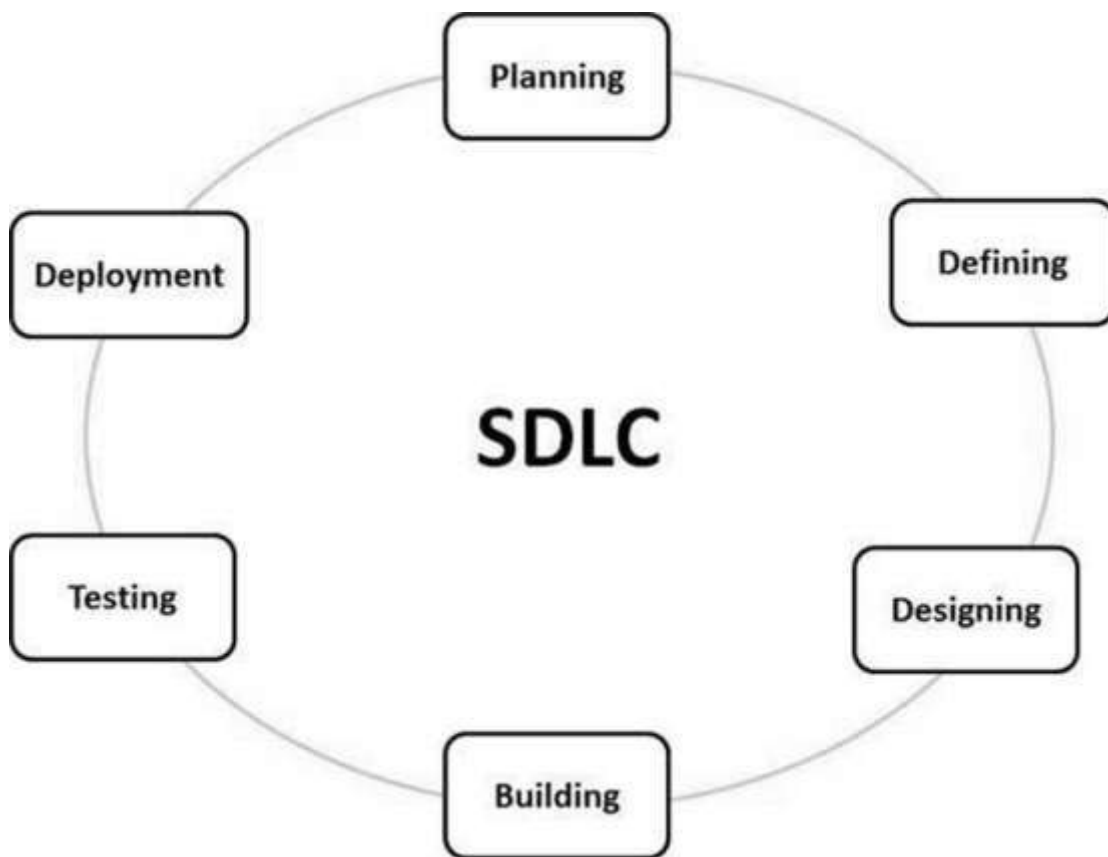
- It is an Android only interface which is used to serialize a class so its properties can be transferred from one activity to another.
- To use it, Java class should be implemented using parcelable interface.
- The parcelable interface adds methods to the class you want to transfer
- It handles the process of copying your object data into a parcel for transmission between activities and then re-creating the object on the other side.

4. What is the difference in an implicit intent and an explicit intent?

<pre>Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url)); startActivity(intent);</pre>	<pre>Intent showDetail = new Intent(this, DetaiActivy.class); startActivity(showDetail);</pre>
Used when you don't know which activity of which applications can handle your request.	Used when you know exactly which Activity can handle your request.
do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it. For example, if you want to show the user a location on a map, you can use an implicit intent to request that another capable app show a specified location on a map.	specify the component to start by name (the fully-qualified class name). You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start.

5. What is the standard software development lifecycle (SDLC)?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.



Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification. This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product. A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle. Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

Models

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model
- Agile Model
- RAD Model
- Rapid Application Development Model
- Prototyping Models