

SESSION 13 & 14

1. What is the purpose of “RecyclerView” in Android?
 - a) Handle background tasks
 - b) Store app data
 - c) Manage user preferences
 - d) Display large datasets efficiently in a scrollable list**

Answer: d

Explanation: RecyclerView is a UI widget used to display large sets of data in a scrollable list format. It efficiently recycles views as they scroll off-screen, ensuring that memory usage remains optimized.

2. Which of the following is NOT part of Android Architecture Components?
 - a) RecyclerView
 - b) LiveData
 - c) ViewModel
 - d) Room**

Answer: a

Explanation: RecyclerView is a UI component for displaying lists and grids efficiently but is not part of the Android Architecture Components. Core components include ViewModel, LiveData, Room, and WorkManager.

3. Why is the view pager used?
 - A) Paging down list items
 - B) Swiping Fragments**
 - C) Swiping Activities
 - D) None

Answer: B

Explanation: ViewPager is used for swiping fragments.

4. All layout classes are subclasses of which of the following?
 - A) android.widget
 - B) android.view.View
 - C) android.view.ViewGroup**
 - D) None

Answer: C

Explanation: All layout classes are subclasses of android.view.ViewGroup.

5. In android, mini activities are also known as
 - A) Adapter
 - B) Activity
 - C) Fragments**
 - D) None

Answer: C

Explanation: In android, mini activities are also known as Fragments.

6. What is a RecyclerView in Android?
 - a) A memory optimization tool
 - b) A layout manager for displaying large data sets**
 - c) A tool for recycling old app code

d) A component for background processing

Answer: b

Explanation: RecyclerView is a more advanced and flexible version of ListView. It is a container for displaying large data sets that can be scrolled very efficiently by maintaining a limited number of views.

7. What is a View in Android?

a) A data storage unit

b) A basic building block for user interface components

c) A memory management tool

d) A type of service

Answer: b

Explanation: In Android, a View is the basic building block for UI components. A View occupies a rectangular area on the screen and is responsible for drawing and event handling.

8. What is the use of the 'findViewById' method in Android?

a) To find data in a database

b) To locate a View in the layout

c) To find an application on the device

d) To search for a file in the file system

Answer: b

Explanation: The 'findViewById' method is used to find and reference views in your activity layout. It returns a view object, allowing you to interact with and modify those views.

9. What is the main purpose of the 'LayoutInflater' in Android?

a) To inflate a new layout from XML

b) To manage app resources

c) To optimize app performance

d) To handle network requests

Answer: a

Explanation: LayoutInflater is used in Android to instantiate the contents of layout XML files into their corresponding View objects.

10. What is 'ViewModel' in Android?

a) A tool for creating 3D models

b) A class designed to store and manage UI-related data

c) A feature for modeling virtual environments

d) A component for managing app views

Answer: b

Explanation: ViewModel is a class that's responsible for preparing and managing the data for an Activity or a Fragment. It also handles the communication of the Activity / Fragment with the rest of the application.

11. How does RecyclerView display items? (Select all that apply)

A) Displays items in a list or a grid.

B) Scrolls vertically or horizontally.

C) Scrolls diagonally on larger devices such as tablets.

D) Allows custom layouts when a list or grid is not enough for the use case.

Explanation: RecyclerView is highly flexible and can display lists, grids, or custom layouts. It supports both vertical and horizontal scrolling but not diagonal scrolling by default.

12. What are the benefits of using RecyclerView? (Select all that apply)
- A) Efficiently displays large lists.
 - B) Automatically updates the data. X
 - C) Minimizes the need for refreshes when an item is updated, deleted, or added to the list.
 - D) Reuses the view when scrolling to automatically display the next item on screen.

Explanation: RecyclerView efficiently handles large datasets by recycling views and minimizing refreshes. However, data changes must be handled manually via the adapter or utilities like DiffUtil.

13. What are some of the reasons for using adapters? (Select all that apply)
- A) Separation of concerns makes it easier to change and test code.
 - B) RecyclerView is agnostic to the data that is being displayed.
 - C) Data processing layers do not have to concern themselves with how data will be displayed.
 - D) The app will run faster. X
- Explanation:** Adapters separate data handling from view management, making the code modular and testable. While RecyclerView can improve performance, using adapters doesn't directly make the app faster unless views are efficiently managed.

14. Which of the following are true of ViewHolder? (Select all that apply)
- A) The ViewHolder layout is defined in XML layout files. X
 - B) There is one ViewHolder for each unit of data in the dataset. X
 - C) You can have more than one ViewHolder in a RecyclerView.
 - D) The Adapter binds data to the ViewHolder.
- Explanation:** A ViewHolder is a Java/Kotlin class, not an XML layout. A ViewHolder represents an item in the dataset, but multiple data items can reuse the same ViewHolder class. You can define multiple ViewHolder types for different layouts like headers, footers, or data items.

15. Which of the following are necessary to use DiffUtil? (Select all that apply)
- A) Extend the ItemCallback class.
 - B) Override areItemsTheSame().
 - C) Override areContentsTheSame().
 - D) Use data binding to track the differences between items. X
- Explanation:** DiffUtil helps efficiently detect changes in a dataset. It requires a custom implementation of ItemCallback, with areItemsTheSame() and areContentsTheSame() methods. Data binding is optional.

16. Which of the following are true about binding adapters? (Select all that apply)
- A) A binding Adapter is a function annotated with @BindingAdapter.
 - B) Using a binding Adapter allows you to separate data formatting from the view holder.
 - C) You must use a RecyclerViewAdapter if you want to use binding adapters. X
 - D) Binding adapters are a good solution when you need to transform complex data.

Explanation: Binding adapters allow you to define custom logic for data binding in XML, separate from the ViewHolder. They are useful for complex data transformations but are not limited to RecyclerViews.

17. Which of the following are layout managers provided by Android? (Select all that apply)

- A) LinearLayoutManager
- B) GridLayoutManager
- C) CircularLayoutManager
- D) StaggeredGridLayoutManager

Explanation: Android provides built-in LinearLayoutManager, GridLayoutManager, and StaggeredGridLayoutManager. CircularLayoutManager is not a built-in option.

18. What is a span in GridLayoutManager?

- A) The size of a grid created by GridLayoutManager.
- B) The width of a column in the grid.
- C) The dimensions of an item in a grid.
- D) The number of columns in a grid that has a vertical orientation.

Explanation: In GridLayoutManager, a span defines how many columns (or rows in horizontal orientation) a grid will have.

19. Where do you add the android:onClick attribute to make items in a RecyclerView respond to clicks? (Select all that apply)

- A) In the layout file that displays the RecyclerView, add it to <androidx.recyclerview.widget.RecyclerView>
- B) Add it to the layout file for an item in the row. If you want the entire item to be clickable, add it to the parent view that contains the items in the row.
- C) Add it to the layout file for an item in the row. If you want a single TextView in the item to be clickable, add it to the <TextView>.
- D) Always add it to the layout file for the MainActivity.

Explanation: Click listeners should be added at the item layout level, not on the RecyclerView itself. You can add it to specific views inside an item.

20. Which of the following statements is true about ViewHolder?

- A) An Adapter can use multiple ViewHolder classes to hold headers and various types of data.
- B) You can have exactly one view holder for data, and one view holder for a header.
- C) A RecyclerView supports multiple types of headers, but the data has to be uniform.
- D) When adding a header, you subclass RecyclerView to insert the header at the correct position.

Explanation: You can have multiple ViewHolder classes for different data types like headers, data items, and footers. You don't subclass RecyclerView to add headers — you handle it in the adapter logic.

21. Which of the following best describes how RecyclerView optimizes performance compared to ListView?

- A) RecyclerView preloads all items in the list, improving performance.
- B) RecyclerView automatically creates a separate thread for each item.

- C) RecyclerView reuses item views via ViewHolder pattern to minimize resource usage.
- D) RecyclerView binds all data items at once for faster loading.

Answer: C

Explanation: RecyclerView improves performance by reusing view holders, reducing the number of view creations during scrolling.

22. What is one key feature of RecyclerView not available in ListView?

- A) Support for heterogeneous layouts and multiple view types in a single list.**
- B) Ability to display a scrollable list of items.
- C) Adapter-based data binding for views.
- D) Click handling on individual items.

Answer: A

Explanation: RecyclerView supports multiple item view types (e.g., headers, footers, different layouts), which ListView does not handle as easily.

23. Why is RecyclerView considered more flexible than ListView?

- A) RecyclerView has built-in support for drag and drop and swipe gestures.**
- B) RecyclerView automatically manages item heights.
- C) RecyclerView doesn't require an adapter for data binding.
- D) RecyclerView uses more memory for better performance.

Answer: A

Explanation: RecyclerView provides built-in support for complex interactions like drag and drop or swipe-to-delete.

24. Which layout manager is not directly supported by ListView but is supported by RecyclerView?

- A) LinearLayoutManager
- B) GridLayoutManager
- C) StaggeredGridLayoutManager
- D) All of the above**

Answer: D

Explanation: RecyclerView supports multiple layout managers like Linear, Grid, and StaggeredGrid, while ListView is limited to a vertical list.

25. Which of the following is true about ListView vs RecyclerView?

- A) ListView supports smooth animations for item changes without extra work.
- B) RecyclerView has better support for layout animations and item updates.**
- C) ListView requires a separate ViewHolder pattern implementation.
- D) RecyclerView does not support scrolling.

Answer: B

Explanation: RecyclerView has built-in support for animations like item addition, removal, and updates.

26. What is the primary role of the RecyclerView.Adapter class?

- A) It creates the layout for each item in the RecyclerView.
- B) It binds data to views and creates ViewHolders.**
- C) It manages click events for each item in the list.
- D) It defines the layout manager for the RecyclerView.

Answer: B

Explanation: Adapter connects the data source to the RecyclerView by binding data to the ViewHolders.

27. When should onCreateViewHolder() be called in the RecyclerView.Adapter lifecycle?

- A) Every time a new item is scrolled into view.
- B) When a new ViewHolder needs to be created.**
- C) When data is updated in the list.
- D) When the RecyclerView is first initialized.

Answer: B

Explanation: onCreateViewHolder() is called to create a new ViewHolder when no existing view can be reused.

28. Which method in RecyclerView.Adapter is used to update the content of a ViewHolder?

- A) onBindViewHolder()**
- B) onCreateViewHolder()
- C) getItemViewType()
- D) getItemCount()

Answer: A

Explanation: onBindViewHolder() binds data from the dataset to the ViewHolder's views.

29. How does RecyclerView.Adapter determine the total number of items to display?

- A) It calls the size of the data list in the constructor.
- B) It uses the return value of getItemCount().**
- C) It checks the layout manager's configuration.
- D) It automatically calculates it based on the view type.

Answer: B

Explanation: The adapter's getItemCount() method tells RecyclerView how many items to display.

30. What is the purpose of the RecyclerView.ViewHolder class?

- A) It holds references to the views for a data item, improving performance.**
- B) It is responsible for creating the entire RecyclerView layout.
- C) It handles click events for the RecyclerView.
- D) It replaces the need for an adapter in RecyclerView.

Answer: A

Explanation: ViewHolder stores references to item views, reducing calls to findViewById and improving performance.

31. Which of the following is a correct way to create a custom ViewHolder in RecyclerView?

- A) Extend RecyclerView.Adapter directly.
- B) Extend RecyclerView.ViewHolder and bind views inside the constructor.**
- C) Add view references inside the RecyclerView layout.
- D) Use onBindViewHolder() to create new views every time.

Answer: B

Explanation: The proper way is to create a custom ViewHolder class extending RecyclerView.ViewHolder and initialize view references in its constructor.

32. What parameter is passed to a ViewHolder's constructor?

- A) The entire data list.
- B) A reference to the parent RecyclerView.
- C) The inflated itemView layout.**
- D) The Adapter's position.

Answer: C

Explanation: The constructor of a ViewHolder receives the inflated view representing a single item.

33. Which is the best approach to handle clicks on items in a RecyclerView?

- A) Set an OnClickListener on the RecyclerView itself.
- B) Use the android:onClick attribute in the RecyclerView layout.
- C) Set an OnClickListener inside the ViewHolder constructor.**
- D) Use ListView's OnItemClickListener.

Answer: C

Explanation: The recommended way is to set the OnClickListener inside the ViewHolder to handle clicks for each item.

34. How can you pass the position of an item to a click listener in a ViewHolder?

- A) By using getItemCount() method.
- B) By hardcoding the position in the click listener.
- C) By calling getAdapterPosition() inside the ViewHolder.**
- D) By adding an extra field in the ViewHolder class.

Answer: C

Explanation: getAdapterPosition() returns the current position of the item in the adapter.

35. What is a key reason to handle clicks inside the ViewHolder rather than in the Adapter?

- A) It keeps the Adapter lightweight and reusable.**
- B) It allows multiple click listeners in the RecyclerView.
- C) It makes the ViewHolder act as an Activity.
- D) It is not possible to handle clicks in the ViewHolder.

Answer: A

Explanation: Handling clicks in the ViewHolder keeps the adapter focused on binding data, improvising modularity and reusability.

SESSION15

1. ActionBar can be associated to

- (a) Only Fragments
- (b) Only Activities**
- (c) Both Activities And Fragments
- (d) None Of The Above

Answer:

Option **(b)**

2. Button option can be choose from the palette category

- (a) textfields
- (b) containers**

- (c) widgets
- (d) layouts

Answer:

Option (c)

3. Which of the following are UI elements that you can use in a window in an Android application?
 - (a) TextBox
 - (b) TextView
 - (c) EditText
 - (d) Both B & C**

Answer:

Option (d)

4. _____ views are commonly used views such as the TextView, EditText and Button Views
 - (a) Basic View**
 - (b) Picker view
 - (c) Frame view
 - (d) None

Answer:

Option (a)

5. The _____ attribute is an identifier for a view so that it may later be retrieved using the View.findViewById method.
 - (a) id**
 - (b) X axis
 - (c) Y axis
 - (d) None

Answer:

Option (a)

6. _____ is a view group that displays child views in relative positions
 - (a) Table layout
 - (b) Linear layout
 - (c) Relative layout**
 - (d) None

Answer:

Option (c)

7. What is the recommended way to handle item click events from an ActionBar menu in an Activity?
 - (a) Overriding onClick() of the menu item
 - (b) Setting click listener to each MenuItem object
 - (c) Overriding onOptionsItemSelected(MenuItem item)**
 - (d) Implementing MenuItem.OnClick Listener in Activity

Answer:

c) Overriding onOptionsItemSelected(MenuItem item)

8. Which method is used to add ActionBar items defined in a menu resource?
- (a) addMenuItem(R.menu.item_menu)
 - (b) inflate(R.menu.item_menu, this)
 - (c) onCreateOptionsMenu(Menu menu) and getMenuInflater().inflate(...)**
 - (d) onCreateActionMenu(Menu menu)

Answer:

c) onCreateOptionsMenu(Menu menu) and getMenuInflater().inflate(...)

9. When replacing the ActionBar with a ToolBar, what must be done in the Activity code?
- (a) Set the theme to NoActionBar and use setSupportActionBar()**
 - (b) Add ToolBar in layout and call setSupportActionBar()
 - (c) Use ToolBar only inside fragments
 - (d) Add ToolBar in layout without any additional code

Answer:

a) Set the theme to NoActionBar and use setSupportActionBar()

10. What is the key difference between ActionBar and ToolBar in Android?
- (a) ToolBar only works in API 21 and above
 - (b) ToolBar cannot be customized
 - (c) ToolBar is a View, ActionBar is not**
 - (d) ActionBar provides better performance

Answer:

c) ToolBar is a View, ActionBar is not

11. What distinguishes a PopupMenu from a Contextual Action Mode in Android?
- (a) PopupMenu is displayed at the top, Contextual Mode is at the bottom
 - (b) Contextual Mode takes over the ActionBar; PopupMenu doesn't**
 - (c) PopupMenu allows multiple item selection
 - (d) Contextual Action Mode supports dialogs

Answer:

b) Contextual Mode takes over the ActionBar; PopupMenu doesn't

12. Which method is used to display a PopupWindow after inflation?
- (a) popupWindow.create()
 - (b) popupWindow.show()
 - (c) popupWindow.showAtLocation()**
 - (d) popupWindow.render()

Answer:

c) popupWindow.showAtLocation()

13. Which of the following is a correct combination for implementing tabbed UI with fragments?
- (a) ViewPager2 + ViewModel
 - (b) ViewPager + FragmentManager + TabLayout**
 - (c) ViewPager + LinearLayout + RecyclerView
 - (d) ViewPager + ConstraintLayout + ListView

Answer:

b) ViewPager + FragmentManager + TabLayout

14. What is the purpose of FragmentStatePagerAdapter over FragmentPagerAdapter?

- (a) It stores more fragment state in memory
- (b) It prevents state saving of fragments
- (c) It destroys off-screen fragments to save memory**
- (d) It loads all fragments at once

Answer:

c) It destroys off-screen fragments to save memory

15. Which layout is typically used as the root container when implementing a Navigation Drawer?

- (a) LinearLayout
- (b) DrawerLayout**
- (c) FrameLayout
- (d) RelativeLayout

Answer:

b) DrawerLayout

16. What method is used to open the Navigation Drawer programmatically?

- (a) drawer.open()
- (b) openDrawer(GravityCompat.START)**
- (c) toggleDrawer(true)
- (d) showDrawer(Gravity.LEFT)

Answer:

b) openDrawer(GravityCompat.START)

17. What is the use of ActionBarDrawerToggle in a Navigation Drawer setup?

- (a) It toggles between ActionBar and ToolBar
- (b) It synchronizes drawer icon with drawer state**
- (c) It disables drawer gestures
- (d) It locks the drawer in closed position

Answer:

b) It synchronizes drawer icon with drawer state

18. What should be done to maintain Navigation Drawer state across configuration changes like rotation?

- (a) Save the drawer view state in SharedPreferences
- (b) Use setRetainInstance(true) on drawer fragment
- (c) Save drawer state in onSaveInstanceState()
- (d) Add android:configChanges="orientation|screenSize" in manifest**

Answer:

d) Add android:configChanges="orientation|screenSize" in manifest

19. What method is used to inflate a menu resource inside the onCreateOptionsMenu() callback?

- (a) inflateMenu()
- (b) getMenuInflater().inflate()**

- (c) setMenu()
- (d) addMenu()

Answer:

b) getMenuInflater().inflate()

20. Which attribute in Toolbar controls the title text color?

- (a) app:contentInsetStart
- (b) android:titleColor
- (c) app:titleTextColor**
- (d) android:textColor

Answer:

c) app:titleTextColor

21. How can you add a navigation icon (hamburger icon) to a Toolbar?

- (a) setNavigationIcon() method
- (b) addIcon() method
- (c) setNavigationIcon() method**
- (d) setNavIcon() method

Answer:

c) setNavigationIcon() method

22. Which interface is implemented to handle navigation item clicks in NavigationView?

- (a) NavigationView.OnClickListener
- (b) NavigationView.OnItemSelectedListener
- (c) NavigationView.OnNavigationItemSelectedListener**
- (d) NavigationItemClickListener

Answer:

c) NavigationView.OnNavigationItemSelectedListener

23. What is the main advantage of Toolbar over ActionBar?

- (a) Toolbar is faster to load
- (b) Toolbar provides default animations
- (c) Toolbar is more customizable and flexible**
- (d) Toolbar uses less memory

Answer:

c) Toolbar is more customizable and flexible

SESSION16

SESSION16

1. The first step when working with SQLite is to create a class that inherits from a helper class, what is it?

- (a) SQLiteOpenHelper class**
- (b) SQLiteHelper class
- (c) SQLiteDatabaseHelper class
- (d) SQLiteDatabase class

Answer: **(a) SQLiteOpenHelper class**

2. Name the method that enables you to obtain the path of the external storage of an Android device.
- (a) **getExternalStorageDirectory()**
 - (b) `getPathExternalStorageDirectory()`
 - (c) `getExternalStorageFile()`
 - (d) `getExternalStoragePath()`
- Answer: **(a) getExternalStorageDirectory()**
3. "There are two statements: Statement A: Using Shared Preferences, you can store private primitive data only. Statement B: Using External Storage option, you can store public data on the shared external storage. Which of them are correct?"
- (a) Statement A is False, and Statement B is False
 - (b) **Statement A is True, and Statement B is True**
 - (c) Statement A is True, and Statement B is False
 - (d) Statement A is False, and Statement B is True
- Answer: **(b) Statement A is True, and Statement B is True**
4. In Android, by default, SQLite database save data in?
- (a) Memory
 - (b) External Storage
 - (c) **Internal Storage**
 - (d) On the cloud
- Answer: **(c) Internal Storage**
5. Which database is natively supported by Android?
- (a) MSSQL
 - (b) MySQL
 - (c) **SQLite**
 - (d) Oracle
- Answer: **(c) SQLite**
6. On Android SQLite where can we define tables, columns, views, triggers?
- (a) `onUpgrade()`
 - (b) `onInit()`
 - (c) `onStart()`
 - (d) **onCreate()**
- Answer: **(d) onCreate()**
7. How do we get access to the preference?
- (a) Via `getPreference()` method
 - (b) Via `getSharedPreference()` method
 - (c) Via `getDefaultSharedPreferences()` method
 - (d) **All of above**
- Answer: **(d) All of above**
8. There are five different methods to store persistent data. They are: Shared Preference, Internal Storage, External Storage, Network and _____?
- (a) Core Data
 - (b) **SQLite Database**
 - (c) Web Service
- Answer: **(b) SQLite Database**
9. What type of storage is commonly used to store large amounts of structured data in Android?
- (a) Shared Preferences

- (b) External Storage
- (c) Internal Storage
- (d) **SQLite Database**

Answer: **(d) SQLite Database**

10. Which of the following is the default storage location for Android app data?

- (a) Shared Preferences
- (b) SQLite Database
- (c) **Internal storage**
- (d) External storage

Answer: **(c) Internal storage**

11. How can sensitive data like passwords be securely stored in Android?

- (a) Using SharedPreferences
- (b) Embedding in the app's source code
- (c) Storing in plain text
- (d) **Using the Android Keystore System**

Answer: **(d) Using the Android Keystore System**

12. What is the most common file type of media supported for audio playback on the Android?

- (a) midi
- (b) **mp3**
- (c) ogg
- (d) wav

Answer: **(b) mp3**

13. Definition of Loader?

- (a) **loaders make it easy to asynchronously load data in an activity or fragment**
- (b) loaders make it easy to synchronously load data in an activity or fragment
- (c) loaders does not make it easy to asynchronously load data in an activity or fragment
- (d) None of the above

Answer: **(a) loaders make it easy to asynchronously load data in an activity or fragment**

14. How to access the context in android content provider?

- (a) Using getContext() in onCreate()
- (b) Using intent()
- (c) Using getApplicationContext() at anywhere in an application
- (d) **A & C**

Answer: **(d) A & C**

15. How many protection levels are available in the android permission tag?

- (a) There are no permission tags available in android
- (b) **Normal, dangerous, signature, and signatureOrsystem**
- (c) Normal, kernel, application
- (d) Kernel, Normal, Application

Answer: **(b) Normal, dangerous, signature, and signatureOrsystem**

16. ViewPager is used for

- (a) Swiping Activities
- (b) **Swiping Fragments**
- (c) Paging Down List Items
- (d) View Pager Is Not Supported By Android SDK

Answer: **(b) Swiping Fragments**

17. The requests from Content Provider class is handled by method

- (a) **onCreate**

- (b) onSelect
- (c) onClick
- (d) ContentResolver

Answer: **(a) onCreate**

18. A type of service provided by android that helps in creating user interfaces is

- (a) Notifications Manager
- (b) Content Providers
- (c) Activity Manager
- (d) **View System**

Answer: **(d) View System**

19. The android component that works like database

- (a) Services
- (b) Activities
- (c) Broadcast Receivers
- (d) **Content Providers**

Answer: **(d) Content Providers**

20. One of the application component, that manages the database issues is called

- (a) Services
- (b) Broadcast Receivers
- (c) **Content Providers**
- (d) Activities

Answer: **(c) Content Providers**

21. What method is used to register a BroadcastReceiver in code?

- (a) **registerReceiver()**
- (b) registerBroadcast()
- (c) bindReceiver()
- (d) setReceiver()

Answer: **(a) registerReceiver()**

22. Which of the following is not a valid callback method in the activity lifecycle?

- (a) onStart()
- (b) **onRun()**
- (c) onPause()
- (d) onDestroy()

Answer: **(b) onRun()**

23. What is ANR in Android?

- (a) Application Not Running
- (b) Android Not Responding
- (c) **Application Not Responding**
- (d) App Not Rebooting

Answer: **(c) Application Not Responding**

24. What component is used to show brief messages at the bottom of the screen in Android?

- (a) AlertDialog
- (b) Toast
- (c) **Snackbar**
- (d) Notification

Answer: **(c) Snackbar**

25. What method in the Service class is called only once in the service lifecycle?

- (a) onStart()

- (b) onStartCommand()
- (c) **onCreate()**
- (d) onBind()

Answer: **(c) onCreate()**

26. Which manifest element is used to declare a service?

- (a) <activity>
- (b) <receiver>
- (c) <provider>
- (d) **<service>**

Answer: **(d) <service>**

27. Which class is used to create notifications in Android?

- (a) AlertDialog.Builder
- (b) NotificationManager
- (c) Toast
- (d) **NotificationCompat.Builder**

Answer: **(d) NotificationCompat.Builder**

28. Which folder contains the XML layout files in an Android project?

- (a) /src
- (b) /layout
- (c) **/res/layout**
- (d) /res/xml

Answer: **(c) /res/layout**

29. What is the use of AndroidManifest.xml?

- (a) **It declares components of the application**
- (b) It stores user preferences
- (c) It defines UI layout
- (d) It registers Java libraries

Answer: **(a) It declares components of the application**

SESSION 17

1. Which of the following permissions is essential for accessing contacts via ContactsContract.Contacts.CONTENT_URI?

- a) android.permission.READ_EXTERNAL_STORAGE
- b) android.permission.READ_CONTACTS**
- c) android.permission.MANAGE_DOCUMENTS
- d) android.permission.ACCESS_NETWORK_STATE

Answer: **b) android.permission.READ_CONTACTS**

2. You are creating a custom ContentProvider for your app's SQLite database. What method is responsible for inserting new data into the underlying database?

- a) insertData()
- b) onInsert()
- c) insert(Uri uri, ContentValues values)**
- d) onCreateInsert(Uri uri, ContentValues values)

Answer: **c) insert(Uri uri, ContentValues values)**

3. Which URI format correctly refers to a specific contact with ID 4 in a Content Provider?

- a) content://contacts/4
- b) content://com.android.contacts/contacts/4**
- c) content://com.contacts.provider/4
- d) content://contacts.provider/data/4

Answer: b) content://com.android.contacts/contacts/4

4. Which threading approach in Android is recommended for performing **UI updates from background threads** safely?

- a) Thread.sleep()
- b) AsyncTask.doInBackground()
- c) Handler.post(Runnable)**
- d) System.runFinalization()

Answer: c) Handler.post(Runnable)

5. In a multi-threaded Android application, what does Looper.prepare() do before calling Looper.loop() in a custom thread?

- a) It connects the thread with the UI.
- b) It creates a message queue for the thread.**
- c) It blocks the thread until the UI finishes.
- d) It terminates the thread gracefully.

Answer: b) It creates a message queue for the thread.

6. When should you prefer using AsyncTask over Thread?

- a) For continuous long-running background processes
- b) For frequent data syncs in services
- c) For short tasks needing result back on the UI thread**
- d) When managing multiple database writes

Answer: c) For short tasks needing result back on the UI thread

7. Which method of AsyncTask runs on the **UI thread**?

- a) doInBackground()
- b) run()
- c) execute()
- d) onPostExecute()**

Answer: d) onPostExecute()

8. You have a custom ContentProvider. Which method must you override to support querying specific rows with a WHERE clause?

- a) onQuery(Uri, Projection)
- b) query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)**
- c) getType(Uri uri)

d) resolveQuery()

Answer: b) query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)

9. In Android, if a background thread tries to directly modify the UI, what will happen?

a) UI will automatically update

b) It will throw RuntimeException

c) The app will crash silently

d) Nothing will happen, it's allowed

Answer: b) It will throw RuntimeException

10. What is the **correct sequence** of AsyncTask method calls?

a) doInBackground() → onPostExecute() → onPreExecute()

b) onPreExecute() → doInBackground() → onPostExecute()

c) doInBackground() → onPostExecute() → onPreExecute()

d) execute() → onPostExecute() → doInBackground()

Answer: b) onPreExecute() → doInBackground() → onPostExecute()

Content Providers

1. What component enables data sharing between apps securely and in a structured way?

a) SharedPreferences

b) SQLiteDatabase

c) ContentProvider

d) IntentService

Answer: c) ContentProvider

2. Which of the following MIME types is used for a directory of items in a Content Provider?

a) vnd.android.cursor.item/vnd.<authority>.<path>

b) vnd.android.cursor.dir/vnd.<authority>.<path>

c) text/vnd.android.cursor

d) vnd.android.provider.item/path

Answer: b) vnd.android.cursor.dir/vnd.<authority>.<path>

3. What does the getType(Uri uri) method in a ContentProvider return?

a) MIME type

b) Path string

c) JSON object

d) SQL query

Answer: a) MIME type

4. Which ContentResolver method is used to add data to a ContentProvider?

- a) resolve()
- b) pushData()
- c) insert(Uri, ContentValues)
- d) add(Uri, Cursor)

Answer: c) insert(Uri, ContentValues)

5. What URI structure is typically used to access all rows in a custom content provider?

- a) content://authority/table_name
- b) http://authority/table_name
- c) content:/authority/table_name
- d) content://authority/path/*

Answer: a) content://authority/table_name

6. What happens if you don't implement getType() in your custom content provider?

- a) Compilation error
- b) App crash when content is accessed externally
- c) Default MIME type is returned
- d) Nothing, it's optional

Answer: b) App crash when content is accessed externally

7. What does the returned Cursor represent?

```
java
CopyEdit
Cursor cursor = getContentResolver().query(
    ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
    null, null, null, null);
```

- a) List of SMS messages
- b) Call logs
- c) Phone numbers of all contacts
- d) Email IDs

Answer: c) Phone numbers of all contacts

8. What role does UriMatcher play in a custom ContentProvider?

- a) Verifies database connection
- b) Maps URIs to operation codes
- c) Encrypts data
- d) Sends data to external storage

Answer: b) Maps URIs to operation codes

9. What will happen in the code below?

```
java
CopyEdit
ContentValues values = new ContentValues();
values.put("name", "Alex");
values.put("age", 25);
Uri uri = getContentResolver().insert(
    Uri.parse("content://com.example.provider/users"), values);
```

- a) It will throw an exception
- b) It inserts a user named Alex
- c) Content provider is not allowed to insert
- d) This format is only for file providers

Answer: b) It inserts a user named Alex

10. Which permission is needed to query call logs using a content provider?

- a) android.permission.READ_CALL_LOG
- b) android.permission.READ_SMS
- c) android.permission.READ_PHONE_NUMBERS
- d) android.permission.ACCESS_LOGS

Answer: a) android.permission.READ_CALL_LOG

11. If your ContentProvider supports bulk inserts, which method should be overridden?

- a) bulkInsert(Uri, ContentValues[])
- b) insertMany()
- c) queryAll()
- d) multiInsert()

Answer: a) bulkInsert(Uri, ContentValues[])

12. Which component below can NOT access a content provider?

- a) Another application
- b) A fragment
- c) An activity
- d) A broadcast receiver

Answer: d) A broadcast receiver

◆ **Multitasking: Threads, Handlers, AsyncTask**

13. Which interface must a class implement to be passed into a Thread constructor?

- a) Handler

b) AsyncRunnable

c) Runnable

d) Looper

Answer: c) Runnable

14. What does t.start() do here?

```
java
CopyEdit
Thread t = new Thread(new Runnable() {
    public void run() {
        Log.d("TAG", "Thread running");
    }
});
t.start();
```

a) Blocks UI thread

b) Runs on UI thread

c) Executes run() on new thread

d) Crashes due to missing Looper

Answer: c) Executes run() on new thread

15. Why is it unsafe to update UI elements directly from a thread?

a) Views are null outside onCreate()

b) UI elements are locked during threads

c) Only main thread owns the View hierarchy

d) Thread class doesn't support UI

Answer: c) Only main thread owns the View hierarchy

16. Which of the following is NOT true about Handler in Android?

a) It can post messages to a message queue

b) It must be bound to a thread with a Looper

c) It blocks the main thread

d) It enables communication between threads

Answer: c) It blocks the main thread

17. What is the result of the below Handler code?

```
java
CopyEdit
Handler handler = new Handler(Looper.getMainLooper());
handler.post(new Runnable() {
    public void run() {
        textView.setText("Updated!");
    }
})
```

}) ;

- a) Crash due to wrong Looper
- b) Updates UI on main thread
- c) Runs on background thread
- d) Doesn't execute

Answer: b) Updates UI on main thread

18. AsyncTask is deprecated because:

- a) It's inefficient for background work
- b) It blocks network usage
- c) It cannot return results
- d) It leaks memory and is lifecycle-unaware

Answer: d) It leaks memory and is lifecycle-unaware

19. What will be printed in the Log?

```
java
CopyEdit
new AsyncTask<Void, Void, String>() {
    protected String doInBackground(Void... params) {
        return "Background Result";
    }

    protected void onPostExecute(String result) {
        Log.d("Result", result);
    }
}.execute();
```

- a) doInBackground
- b) null
- c) Background Result
- d) Nothing, because it throws error

Answer: c) Background Result

20. Which AsyncTask method is used to update progress to the UI thread?

- a) onPreExecute()
- b) onProgressUpdate()
- c) publishProgress()
- d) updateUI()

Answer: c) publishProgress()

21. What Android class is a combination of a background thread + message queue?

- a) Runnable

b) AsyncTask

c) Looper

d) HandlerThread

Answer: d) HandlerThread

22. Which one is the correct use of Runnable?

a) Runnable.run()

b) Runnable.get()

c) Thread(Runnable).start()

d) start(Runnable)

Answer: c) Thread(Runnable).start()

23. What is the result of the below code?

```
java
CopyEdit
new Thread(new Runnable() {
    public void run() {
        System.out.println("Thread A");
    }
}).run();
```

a) New thread is started

b) Nothing happens

c) Runs on the current thread

d) Compiles but throws error

Answer: c) Runs on the current thread

24. Which of these is true about Looper.loop()?

a) It terminates the thread

b) It starts a blocking loop to process messages

c) It kills background tasks

d) It closes the Looper

Answer: b) It starts a blocking loop to process messages

25. Handler's message queue is attached to:

a) AsyncTask

b) Service

c) Looper of the thread

d) RecyclerView

Answer: c) Looper of the thread

26. Which method starts the execution of an AsyncTask?

- a) doInBackground()
- b) start()
- c) execute()
- d) onPostExecute()

Answer: c) execute()

27. Which lifecycle issue can cause memory leaks with AsyncTask?

- a) Running it inside a BroadcastReceiver
- b) Not calling cancel() in onDestroy()
- c) Updating database on UI thread
- d) Using Handler on background thread

Answer: b) Not calling cancel() in onDestroy()

28. You have an app with lots of small background tasks. Which is the most modern solution?

- a) Multiple AsyncTasks
- b) HandlerThread
- c) Executors + Runnable
- d) Thread.sleep()

Answer: c) Executors + Runnable

29. What is the role of Looper.prepare()?

- a) Blocks the UI thread
- b) Sets the thread priority
- c) Initializes the message queue for a thread
- d) Starts new thread

Answer: c) Initializes the message queue for a thread

30. If you want to periodically perform a task in background using Handler, what method do you use?

- a) postDelayed()
- b) execute()
- c) runAsync()
- d) sendEmptyMessage()

Answer: a) postDelayed()

SESSION18&19

1. What is the method used to start a Started Service?

- (a) startService()
- (b) bindService()

- (c) onStartService()
- (d) createService()

Answer: (a) startService()

Explanation: This method tells the system to start the service immediately.

2. Which method is called when the service is created for the first time?

- (a) onStart()
- (b) onStartCommand()
- (c) onCreate()
- (d) onBind()

Answer: (c) onCreate()

Explanation: onCreate() is called once when the service is created.

3. What is the method used to bind a **Bounded Service?**

- (a) bindService()
- (b) startService()
- (c) attachService()
- (d) onBindService()

Answer: (a) bindService()

Explanation: This binds the client to the service for IPC.

4. Which method is used to unbind from a service?

- (a) detachService()
- (b) stopService()
- (c) unbindService()
- (d) onServiceDisconnected()

Answer: (c) unbindService()

Explanation: This disconnects the client from the bound service.

5. Which method is not a part of the **Service Lifecycle?**

- (a) onStartCommand()
- (b) onBind()
- (c) onCreate()
- (d) onClick()

Answer: (d) onClick()

Explanation: onClick() is a UI event method, not related to Service lifecycle.

6. What does onStartCommand() return to indicate the service should be recreated if killed?

- (a) START_NOT_STICKY
- (b) START_STICKY
- (c) START_REDELIVER_INTENT
- (d) START_IMMEDIATE

Answer: (b) START_STICKY

Explanation: This tells the system to restart the service if it is killed.

7. Which class should be extended to create a Bounded Service?

- (a) IntentService
- (b) Service
- (c) Binder
- (d) BroadcastReceiver

Answer: (b) Service

Explanation: Bound services extend Service and implement binding logic.

8. Which method in Service must be overridden to return a Binder for IPC?

- (a) onBind()
- (b) onServiceConnected()
- (c) getBinder()
- (d) returnBinder()

Answer: (a) onBind()

Explanation: onBind() returns an IBinder object for clients.

9. IPC stands for:

- (a) Inter Process Connection
- (b) Inter Processor Communication
- (c) Inter Process Communication
- (d) Internal Private Communication

Answer: (c) Inter Process Communication

Explanation: IPC is communication between different processes.

10. Which class is used in Android for implementing IPC using Binder?

- (a) Messenger
- (b) Handler
- (c) IBinder
- (d) InterfaceService

Answer: (c) IBinder

Explanation: IBinder is the base interface for Binder IPC.

11. What is a main feature of IntentService?

- (a) Runs on main thread
- (b) Uses multiple threads
- (c) Handles asynchronous tasks in background
- (d) Requires manual thread creation

Answer: (c) Handles asynchronous tasks in background

Explanation: IntentService handles tasks sequentially on a worker thread.

12. Which method is called to handle the work inside an **IntentService**?

- (a) `onExecute()`
- (b) `onStartCommand()`
- (c) `onBind()`
- (d) `onHandleIntent()`

Answer: (d) `onHandleIntent()`

Explanation: `onHandleIntent()` runs the code for each intent.

13. What happens after `onHandleIntent()` is completed in IntentService?

- (a) Service restarts
- (b) Service is destroyed automatically
- (c) Activity is launched
- (d) Nothing happens

Answer: (b) Service is destroyed automatically

Explanation: IntentService stops itself after handling all intents.

14. What is the main difference between Service and IntentService?

- (a) IntentService runs on UI thread
- (b) IntentService runs on background thread
- (c) Service cannot handle multiple intents
- (d) IntentService runs indefinitely

Answer: (b) IntentService runs on background thread

Explanation: IntentService automatically runs on a background thread.

15. Which method stops a started service?

- (a) `stop()`
- (b) `closeService()`
- (c) `stopService()`
- (d) `detachService()`

Answer: (c) `stopService()`

Explanation: `stopService()` stops the service started by `startService()`.

16. What is the purpose of a **BroadcastReceiver**?

- (a) Display UI
- (b) Send HTTP requests
- (c) Receive and respond to system/app events
- (d) Run database queries

Answer: (c) Receive and respond to system/app events

Explanation: BroadcastReceivers listen to broadcasts like system events.

17. Which method must be overridden in a BroadcastReceiver?

- (a) onCreate()
- (b) onBroadcast()
- (c) onReceive()
- (d) onSend()

Answer: (c) onReceive()

Explanation: onReceive() handles the incoming broadcast.

18. What is required to register a **BroadcastReceiver** in code?

- (a) IntentFilter
- (b) Bundle
- (c) ServiceIntent
- (d) PendingIntent

Answer: (a) IntentFilter

Explanation: IntentFilter specifies which broadcasts receiver listens to.

19. Which system event can be caught using a BroadcastReceiver?

- (a) SMS Received
- (b) App Installed
- (c) Boot Completed
- (d) All of the above

Answer: (d) All of the above

Explanation: BroadcastReceiver can listen to many system events.

20. Which permission is needed to receive BOOT_COMPLETED?

- (a) android.permission.RECEIVE_BOOT
- (b) android.permission.BOOT_EVENT
- (c) android.permission.RECEIVE_BOOT_COMPLETED
- (d) android.permission.BROADCAST_BOOT

Answer: (c) android.permission.RECEIVE_BOOT_COMPLETED

Explanation: This permission allows receiving boot completion broadcast.

21. Custom broadcasts are sent using:

- (a) sendBroadcast(Intent)
- (b) launchActivity(Intent)
- (c) runBroadcast(Intent)
- (d) startReceiver(Intent)

Answer: (a) sendBroadcast(Intent)

Explanation: This method sends a broadcast intent.

22. How can we stop a running **Bounded Service**?

- (a) stopService()

- (b) unbindService()
- (c) killService()
- (d) endService()

Answer: (b) unbindService()

Explanation: Unbinding disconnects clients and can stop the service.

23. A Bounded Service runs as long as:

- (a) Memory is available
- (b) Client is bound to it
- (c) Activity is not destroyed
- (d) App is in background

Answer: (b) Client is bound to it

Explanation: Service lives while clients are bound.

24. Which one is a correct way to start an IntentService?

- (a) new IntentService().start()
- (b) startService(new Intent(this, MyIntentService.class))
- (c) bindService(new Intent(), ...)
- (d) use startThread()

Answer: (b) startService(new Intent(this, MyIntentService.class))

Explanation: IntentService is started like a normal service with intent.

25. Which thread does a BroadcastReceiver run on?

- (a) UI Thread
- (b) Background Thread
- (c) Looper Thread
- (d) Handler Thread

Answer: (a) UI Thread

Explanation: onReceive() runs on the main/UI thread.

26. What happens if a long task is executed inside onReceive() of a BroadcastReceiver?

- (a) Task completes normally
- (b) Receiver automatically spawns thread
- (c) ANR (App Not Responding) may occur
- (d) Receiver stops the task

Answer: (c) ANR (App Not Responding) may occur

Explanation: Long tasks block the main thread causing ANR.

27. Which statement is true about services?

- (a) Service has a UI
- (b) Services run only when Activity is active
- (c) Service runs in the background without UI

- (d) Service can only run when Broadcast is sent

Answer: (c) Service runs in the background without UI

Explanation: Services work without direct user interaction.

28. Which method is used by activity to communicate with a service using Binder?

- (a) startBinder()
- (b) getServiceInstance()
- (c) onBind()
- (d) BinderClass.getService()

Answer: (c) onBind()

Explanation: onBind() returns the Binder interface for communication.

29. Which broadcast cannot be received by a manifest-registered receiver in Android 8+?

- (a) Custom app broadcasts
- (b) BOOT_COMPLETED
- (c) SMS_RECEIVED
- (d) Implicit broadcasts

Answer: (d) Implicit broadcasts

Explanation: Android 8+ restricts manifest receivers for implicit broadcasts.

30. How to receive custom app broadcasts in another app?

- (a) Add <service> in manifest
- (b) Use explicit intent
- (c) Register the receiver with correct IntentFilter
- (d) Create AIDL interface

Answer: (c) Register the receiver with correct IntentFilter

Explanation: Receiver must be registered to listen for custom broadcasts.

31. What is the purpose of a Service in Android?

- (a) To provide UI to the user
- (b) To run background tasks without UI
- (c) To handle user clicks
- (d) To store app preferences

Answer: (b) To run background tasks without UI

Explanation: Services perform long-running operations in background.

32. Which method starts a Service in Android?

- (a) startActivityForResult()
- (b) startService()
- (c) startReceiver()
- (d) startThread()

Answer: (b) startService()

Explanation: This method starts a service.

33. What is the first method called when a service is started?

- (a) onStart()
- (b) onStartService()
- (c) onStartCommand()
- (d) onCreate()

Answer: (d) onCreate()

Explanation: onCreate() is called before onStartCommand().

34. How do you stop a started service?

- (a) stopReceiver()
- (b) stopService()
- (c) finish()
- (d) cancelService()

Answer: (b) stopService()

Explanation: Stops the service started by startService().

35. What is a bound service mostly used for?

- (a) One-time tasks
- (b) UI display
- (c) Long-running operations
- (d) Ongoing client-server communication

Answer: (d) Ongoing client-server communication

Explanation: Bound services provide interface for clients to interact.

36. Which method is mandatory to override when creating a Service?

- (a) onPause()
- (b) onStop()
- (c) onBind()
- (d) onReceive()

Answer: (c) onBind()

Explanation: Must override onBind() to return binder or null.

37. Which Android component listens for broadcast messages?

- (a) Activity
- (b) Service
- (c) BroadcastReceiver
- (d) ViewModel

Answer: (c) BroadcastReceiver

Explanation: BroadcastReceiver handles system and custom broadcasts.

38. Which method is overridden in a BroadcastReceiver?

- (a) onStartCommand()
- (b) onBind()
- (c) onReceive()
- (d) onCreate()

Answer: (c) onReceive()

Explanation: onReceive() is called when broadcast is received.

39. How can you register a BroadcastReceiver dynamically?

- (a) In AndroidManifest.xml
- (b) Using registerReceiver() in code
- (c) Using startService()
- (d) Using IntentFilter only

Answer: (b) Using registerReceiver() in code

Explanation: Dynamic receivers are registered at runtime via code.

40. What does IntentService use internally to run background tasks?

- (a) UI Thread
- (b) AsyncTask
- (c) Worker Thread
- (d) Main Thread

Answer: (c) Worker Thread

Explanation: IntentService uses a separate worker thread for tasks.

41. Which method is automatically called in IntentService to handle work?

- (a) run()
- (b) onExecute()
- (c) onStart()
- (d) onHandleIntent()

Answer: (d) onHandleIntent()

Explanation: This method handles the intent in background thread.

42. Which one is a valid lifecycle method for a Service?

- (a) onStartCommand()
- (b) onStartIntent()
- (c) onResume()
- (d) onPause()

Answer: (a) onStartCommand()

Explanation: onStartCommand() handles start requests for services.

43. Which of the following is NOT a type of Android Service?

- (a) Started Service
- (b) Bound Service
- (c) Foreground Service
- (d) Broadcast Service

Answer: (d) Broadcast Service

Explanation: BroadcastReceiver is not a type of Service.

44. What does bindService() return?

- (a) Nothing
- (b) IBinder interface
- (c) BroadcastReceiver object
- (d) A Service object

Answer: (b) IBinder interface

Explanation: Returns IBinder for client-service communication.

45. What is used to define the type of broadcast a receiver listens for?

- (a) Intent
- (b) IntentFilter
- (c) Service
- (d) Thread

Answer: (b) IntentFilter

Explanation: IntentFilter specifies which intents receiver listens to.

46. Which method is used to send a broadcast in Android?

- (a) sendBroadcast()
- (b) broadcastEvent()
- (c) sendMessage()
- (d) triggerBroadcast()

Answer: (a) sendBroadcast()

Explanation: Sends the broadcast intent to registered receivers.

47. What kind of events are typically received by system broadcast receivers?

- (a) Login success
- (b) App error logs
- (c) Battery low, Boot completed
- (d) View inflation

Answer: (c) Battery low, Boot completed

Explanation: System broadcasts notify important device events.

48. Where do you declare a static BroadcastReceiver in Android?

- (a) In MainActivity

- (b) In the Service class
- (c) In AndroidManifest.xml
- (d) In values.xml

Answer: (c) In AndroidManifest.xml

Explanation: Static receivers are declared in the manifest file.

49. What happens if a service is started and not stopped?

- (a) It throws an error
- (b) It automatically stops
- (c) It continues running until the system kills it
- (d) It finishes after 10 seconds

Answer: (c) It continues running until the system kills it

Explanation: Started services run until explicitly stopped or killed.

50. Can a BroadcastReceiver show a Toast message?

- (a) No
- (b) Yes
- (c) Only in debug mode
- (d) Only if running in the main thread

Answer: (b) Yes

Explanation: BroadcastReceiver runs on the main thread, so can show Toast.

51. Can a bound service interact directly with the activity?

- (a) No
- (b) Yes, using a Binder
- (c) Yes, using a Broadcast
- (d) Yes, using an AsyncTask

Answer: (b) Yes, using a Binder

Explanation: Binder facilitates direct method calls between service and activity.

52. Which of these statements is true for BroadcastReceiver?

- (a) It runs in a service
- (b) It shows notification
- (c) It listens to Intent filters and reacts
- (d) It executes in the background thread

Answer: (c) It listens to Intent filters and reacts

Explanation: BroadcastReceivers react to matching broadcast intents.

53. What happens when you send a broadcast but no receiver is registered?

- (a) App crashes
- (b) Error is shown
- (c) Nothing happens

(d) Logcat shows warning

Answer: (c) Nothing happens

Explanation: Broadcast is ignored if no receivers are listening.

54. What type of context is passed to a BroadcastReceiver's onReceive()?

(a) Application context

(b) UI context

(c) Service context

(d) Receiver context

Answer: (a) Application context

Explanation: The context is from the application environment.

55. Which of the following is used for communication between Service and Activity?

(a) BroadcastReceiver

(b) AIDL

(c) Binder

(d) All of the above

Answer: (d) All of the above

Explanation: All are valid IPC methods depending on use case.

SESSION20&21

1. What is the recommended API for obtaining location in Android?

(a) LocationManager

(b) FusedLocationProviderClient

(c) TelephonyManager

(d) SensorManager

Answer: (b) FusedLocationProviderClient

Explanation: Google's FusedLocationProviderClient is the preferred API for accurate, battery-efficient location.

2. Which Google API is used for displaying maps in Android apps?

(a) Google Maps API

(b) Google Places API

(c) Google Drive API

(d) Google Analytics API

Answer: (a) Google Maps API

Explanation: Google Maps API provides map display and interactions.

3. Which object is used to mark a location on Google Map?

(a) MarkerOptions

(b) LocationManager

(c) MapMarker

(d) GeoPoint

Answer: (a) MarkerOptions

Explanation: MarkerOptions defines properties for markers on the map.

4. Which method is used to add a marker to the map?

- (a) addMarker()
- (b) addLocation()
- (c) setMarker()
- (d) addPoint()

Answer: (a) addMarker()

Explanation: addMarker() adds a marker with specified options.

5. How can you change the map type to satellite view?

- (a) map.setMapType(GoogleMap.MAP_TYPE_SATELLITE)
- (b) map.setType(SATELLITE)
- (c) map.changeView("satellite")
- (d) map.setView(SATELLITE)

Answer: (a) map.setMapType(GoogleMap.MAP_TYPE_SATELLITE)

Explanation: setMapType() changes the map display style.

6. Which listener is used for receiving regular location updates?

- (a) LocationChangeListener
- (b) LocationListener
- (c) FusedLocationListener
- (d) LocationUpdateListener

Answer: (b) LocationListener

Explanation: LocationListener receives callbacks when location changes.

7. What permission is mandatory for getting location updates?

- (a) android.permission.ACCESS_LOCATION
- (b) android.permission.ACCESS_FINE_LOCATION
- (c) android.permission.INTERNET
- (d) android.permission.LOCATION_UPDATE

Answer: (b) android.permission.ACCESS_FINE_LOCATION

Explanation: Required to get precise location.

8. Which method requests location updates from LocationManager?

- (a) requestLocation()
- (b) startLocationUpdates()
- (c) requestLocationUpdates()
- (d) getLocationUpdates()

Answer: (c) requestLocationUpdates()

Explanation: Requests location updates with specified criteria.

9. Which class is used for location coordinates?

- (a) GeoPoint
- (b) LatLng
- (c) LocationPoint
- (d) GeoCoordinate

Answer: (b) LatLng

Explanation: LatLng represents latitude and longitude on maps.

10. Which API helps combine GPS, WiFi, and cell networks to improve location accuracy?

- (a) LocationManager
- (b) Fused Location Provider
- (c) SensorManager
- (d) TelephonyManager

Answer: (b) Fused Location Provider

Explanation: It fuses multiple sources for better accuracy and efficiency.

Telephony Manager (Call and SMS)

11. What Android class provides information about telephony services?

- (a) PhoneManager
- (b) TelephonyManager
- (c) CallManager
- (d) SmsManager

Answer: (b) TelephonyManager

Explanation: TelephonyManager offers telephony info like network, call state.

12. How to get the device's phone number using TelephonyManager?

- (a) getDeviceNumber()
- (b) getLine1Number()
- (c) getPhoneNumber()
- (d) getNumber()

Answer: (b) getLine1Number()

Explanation: getLine1Number() returns the device's phone number.

13. Which permission is needed to read the phone state?

- (a) android.permission.READ_PHONE_STATE
- (b) android.permission.CALL_PHONE
- (c) android.permission.READ_SMS
- (d) android.permission.ACCESS_NETWORK_STATE

Answer: (a) android.permission.READ_PHONE_STATE

Explanation: Permission needed for TelephonyManager info access.

14. Which class manages sending SMS programmatically?

- (a) TelephonyManager
- (b) SmsManager
- (c) CallManager
- (d) SmsSender

Answer: (b) SmsManager

Explanation: SmsManager handles SMS sending.

15. What method sends a text message using SmsManager?

- (a) sendMessage()
- (b) sendSMS()
- (c) sendTextMessage()
- (d) send()

Answer: (c) sendTextMessage()

Explanation: sendTextMessage() sends SMS to a specified number.

16. Which class is used to manage Bluetooth on Android?

- (a) BluetoothController
- (b) BluetoothAdapter
- (c) BluetoothManager
- (d) BluetoothDevice

Answer: (b) BluetoothAdapter

Explanation: BluetoothAdapter is the entry point to all Bluetooth actions.

17. How to check if Bluetooth is enabled?

- (a) bluetoothAdapter.isEnabled()
- (b) bluetoothAdapter.checkOn()
- (c) bluetoothAdapter.isBluetoothOn()
- (d) bluetoothAdapter.checkEnabled()

Answer: (a) bluetoothAdapter.isEnabled()

Explanation: Returns true if Bluetooth is currently enabled.

18. Which intent action is used to request enabling Bluetooth?

- (a) ACTION_REQUEST_ENABLE
- (b) ACTION_ENABLE_BLUETOOTH
- (c) REQUEST_ENABLE_BLUETOOTH
- (d) ACTION_BLUETOOTH_ON

Answer: (a) ACTION_REQUEST_ENABLE

Explanation: Used to launch system dialog to turn on Bluetooth.

19. How to discover nearby Bluetooth devices?

- (a) startDiscovery()
- (b) discoverDevices()
- (c) findDevices()
- (d) startScanning()

Answer: (a) startDiscovery()

Explanation: Begins discovery of nearby Bluetooth devices.

20. Which permission is required to use Bluetooth scanning on Android 12+?

- (a) BLUETOOTH_SCAN
- (b) BLUETOOTH_DISCOVER
- (c) ACCESS_FINE_LOCATION
- (d) BLUETOOTH_ADMIN

Answer: (a) BLUETOOTH_SCAN

Explanation: New Bluetooth scanning permission introduced in Android 12.

Android Sensors: Proximity Sensor

21. Which Android class is used to access device sensors?

- (a) SensorManager
- (b) SensorAdapter
- (c) SensorController
- (d) SensorListener

Answer: (a) SensorManager

Explanation: Manages all sensors on the device.

22. What type of sensor is a proximity sensor?

- (a) Motion sensor
- (b) Position sensor
- (c) Environmental sensor
- (d) Proximity sensor

Answer: (d) Proximity sensor

Explanation: Detects if an object is near the device.

23. Which sensor type constant represents the proximity sensor?

- (a) TYPE_ACCELEROMETER
- (b) TYPE_PROXIMITY
- (c) TYPE_LIGHT
- (d) TYPE_GYROSCOPE

Answer: (b) TYPE_PROXIMITY

Explanation: Constant representing the proximity sensor.

24. How can you register a listener to the proximity sensor?

- (a) `sensorManager.registerListener(listener, sensor, SensorManager.SENSOR_DELAY_NORMAL)`
- (b) `sensorManager.addListener(listener, sensor)`
- (c) `sensorManager.listen(sensor)`
- (d) `sensorManager.registerProximityListener(listener)`

Answer: (a) `sensorManager.registerListener(listener, sensor,`

`SensorManager.SENSOR_DELAY_NORMAL)`

Explanation: Registers a listener for sensor events with a specified delay.

25. What does the proximity sensor usually detect?

- (a) Distance to nearest object in cm
- (b) Temperature near device
- (c) Humidity around device
- (d) Motion speed

Answer: (a) Distance to nearest object in cm

Explanation: It measures the proximity distance typically in centimeters.

26. What value does a proximity sensor return when something is near?

- (a) Maximum range value
- (b) Minimum range value (close to zero)
- (c) Negative value
- (d) Random value

Answer: (b) Minimum range value (close to zero)

Explanation: Sensor returns a low value when near an object.

27. Which method is called when sensor values change?

- (a) `onSensorChanged()`
- (b) `onValueChanged()`
- (c) `onChange()`
- (d) `sensorUpdate()`

Answer: (a) `onSensorChanged()`

Explanation: Called when sensor detects a change in value.

28. What permission is required to access sensors?

- (a) No special permission needed
- (b) `android.permission.ACCESS_SENSORS`
- (c) `android.permission.BODY_SENSORS`
- (d) `android.permissionSENSOR_ACCESS`

Answer: (a) No special permission needed

Explanation: Sensors like proximity do not require special permissions.

29. Which sensor delay mode provides the fastest updates?

- (a) SENSOR_DELAY_UI
- (b) SENSOR_DELAY_NORMAL
- (c) SENSOR_DELAY_GAME
- (d) SENSOR_DELAY_FASTEST

Answer: (d) SENSOR_DELAY_FASTEST

Explanation: Provides the quickest sensor event updates.

30. How to unregister a sensor listener?

- (a) sensorManager.removeListener(listener)
- (b) sensorManager.unregisterListener(listener)
- (c) sensorManager.stopListener(listener)
- (d) sensorManager.unregisterSensor(listener)

Answer: (b) sensorManager.unregisterListener(listener)

Explanation: Stops receiving sensor updates.

Some Code based mcq

1. What will the following code snippet do?

java

CopyEdit

```
FusedLocationProviderClient fusedLocationClient =  
LocationServices.getFusedLocationProviderClient(context);  
  
fusedLocationClient.getLastLocation()  
  
.addOnSuccessListener(location -> {  
  
    if (location != null) {  
  
        Log.d("Location", "Lat: " + location.getLatitude() + ", Lon: " + location.getLongitude());  
  
    }  
  
});
```

- (a) Request continuous location updates
- (b) Get the last known location once
- (c) Register a location listener for updates
- (d) Start a foreground service for location tracking

Answer: (b) Get the last known location once

Explanation: getLastLocation() fetches the most recent cached location; it does not provide continuous updates.

2. What is the output of the following Bluetooth check code if Bluetooth is off?

java

CopyEdit

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();  
if (bluetoothAdapter == null) {  
    Log.d("BT", "Device does not support Bluetooth");  
} else if (!bluetoothAdapter.isEnabled()) {  
    Log.d("BT", "Bluetooth is disabled");  
} else {  
    Log.d("BT", "Bluetooth is enabled");  
}
```

- (a) Bluetooth is enabled
- (b) Bluetooth is disabled
- (c) Device does not support Bluetooth
- (d) NullPointerException occurs

Answer: (b) Bluetooth is disabled

Explanation: The adapter exists but isEnabled() returns false, indicating Bluetooth is off.

3. What does this sensor listener do?

java

CopyEdit

```
SensorEventListener proximityListener = new SensorEventListener() {  
  
    @Override  
  
    public void onSensorChanged(SensorEvent event) {  
  
        if (event.values[0] < event.sensor.getMaximumRange()) {  
            Log.d("Proximity", "Near");  
        } else {  
            Log.d("Proximity", "Far");  
        }  
    }  
  
    @Override  
  
    public void onAccuracyChanged(Sensor sensor, int accuracy) {}  
};
```

- (a) Detects if an object is near or far from the proximity sensor
- (b) Detects accelerometer motion

- (c) Changes screen brightness
- (d) Detects gyroscope rotation

Answer: (a) Detects if an object is near or far from the proximity sensor

Explanation: The listener compares current value to max range to determine proximity.

4. What does this code do?

java

CopyEdit

```
Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
```

- (a) Automatically enables Bluetooth without user interaction
- (b) Requests user permission to enable Bluetooth
- (c) Disables Bluetooth
- (d) Checks if Bluetooth is enabled

Answer: (b) Requests user permission to enable Bluetooth

Explanation: This intent shows a system dialog asking user to enable Bluetooth.

5. What is wrong with this code snippet to request location updates?

java

CopyEdit

```
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 1,
locationListener);
```

- (a) It requires ACCESS_FINE_LOCATION permission
- (b) It requests too frequent updates and will cause error
- (c) The parameters are invalid
- (d) It does not specify provider

Answer: (a) It requires ACCESS_FINE_LOCATION permission

Explanation: GPS location updates require ACCESS_FINE_LOCATION permission to be granted.

1. Which permission is required to access the device's approximate location?

- (a) ACCESS_FINE_LOCATION
- (b) ACCESS_COARSE_LOCATION
- (c) ACCESS_NETWORK_STATE
- (d) INTERNET

Answer: (b) ACCESS_COARSE_LOCATION

Explanation: Coarse location provides approximate location using network sources.

2. What does getLastLocation() in FusedLocationProviderClient return?

- (a) The current location
- (b) The last known cached location
- (c) It starts location updates
- (d) The GPS location only

Answer: (b) The last known cached location

Explanation: Returns cached location if available, else null.

3. Which class do you use to display a Google Map in Android?

- (a) MapView
- (b) MapFragment
- (c) GoogleMap
- (d) MapActivity

Answer: (b) MapFragment

Explanation: MapFragment is the UI container for Google Maps.

4. What does LocationListener interface provide?

- (a) Start location updates
- (b) Stop location updates
- (c) Callbacks when location changes
- (d) Permission handling

Answer: (c) Callbacks when location changes

Explanation: It receives location updates.

5. Which of these is NOT a map type available in Google Maps?

- (a) Satellite
- (b) Terrain
- (c) StreetView
- (d) Normal

Answer: (c) StreetView

Explanation: StreetView is a separate feature, not a map type.

Telephony Manager (Call and SMS)

6. Which method of TelephonyManager returns the current network operator name?

- (a) getNetworkOperatorName()
- (b) getSimOperatorName()
- (c) getOperatorName()
- (d) getCarrierName()

Answer: (a) getNetworkOperatorName()

Explanation: Returns the current registered network operator name.

7. To send an SMS, which class do you use?

- (a) SmsSender
- (b) SmsManager
- (c) TelephonyManager
- (d) SmsClient

Answer: (b) SmsManager

Explanation: Handles SMS sending.

8. What permission is required to send SMS programmatically?

- (a) SEND_SMS
- (b) READ_SMS
- (c) WRITE_SMS
- (d) RECEIVE_SMS

Answer: (a) SEND_SMS

Explanation: Required to send SMS messages.

9. getLine1Number() returns:

- (a) IMEI number
- (b) Device phone number
- (c) SIM serial number
- (d) Subscriber ID

Answer: (b) Device phone number

Explanation: Returns the phone number of the device.

Bluetooth

10. Which method is used to start Bluetooth device discovery?

- (a) discoverDevices()
- (b) startDiscovery()
- (c) scanDevices()
- (d) findDevices()

Answer: (b) startDiscovery()

Explanation: Begins scanning for nearby Bluetooth devices.

11. Which intent is used to ask the user to enable Bluetooth?

- (a) ACTION_ENABLE_BLUETOOTH
- (b) ACTION_REQUEST_ENABLE
- (c) ACTION_TURN_ON_BLUETOOTH
- (d) ACTION_BLUETOOTH_ON

Answer: (b) ACTION_REQUEST_ENABLE

Explanation: Launches a system dialog to enable Bluetooth.

12. What will BluetoothAdapter.getDefaultAdapter() return if device doesn't support Bluetooth?

- (a) Null
- (b) Empty adapter
- (c) Throws Exception
- (d) A dummy adapter

Answer: (a) Null

Explanation: Returns null if Bluetooth hardware is not available.

13. Which sensor type is used to detect nearby objects without physical contact?

- (a) Accelerometer
- (b) Proximity Sensor
- (c) Gyroscope
- (d) Light Sensor

Answer: (b) Proximity Sensor

Explanation: Detects presence of nearby objects.

14. Which method is called to register a sensor listener?

- (a) registerListener()
- (b) addListener()
- (c) startListening()
- (d) registerSensor()

Answer: (a) registerListener()

Explanation: Registers a listener for sensor events.

15. The proximity sensor usually returns what value when no object is near?

- (a) Zero
- (b) Maximum range value
- (c) Negative value
- (d) Random value

Answer: (b) Maximum range value

Explanation: Indicates sensor sees no nearby object.

16. Which of the following permissions is needed to access the proximity sensor?

- (a) BODY_SENSORS
- (b) No special permission
- (c) SENSOR_ACCESS
- (d) ACCESS_FINE_LOCATION

Answer: (b) No special permission

Explanation: Proximity sensor access requires no special permissions.

Mixed / General

17. What is the main advantage of using FusedLocationProviderClient over LocationManager?

- (a) Uses less battery and more accuracy
- (b) Requires no permissions
- (c) Supports SMS sending
- (d) Works offline only

Answer: (a) Uses less battery and more accuracy

Explanation: Combines multiple sources intelligently.

18. Which API do you use to get SIM card info?

- (a) TelephonyManager
- (b) SmsManager
- (c) BluetoothAdapter
- (d) LocationManager

Answer: (a) TelephonyManager

Explanation: Provides telephony and SIM info.

19. What Bluetooth permission is mandatory for scanning devices on Android 12+?

- (a) BLUETOOTH_SCAN
- (b) BLUETOOTH_CONNECT
- (c) BLUETOOTH_ADMIN
- (d) BLUETOOTH_DISCOVER

Answer: (a) BLUETOOTH_SCAN

Explanation: New runtime permission introduced in Android 12.

20. Which callback method receives sensor data updates?

- (a) onSensorChanged()
- (b) onUpdate()
- (c) sensorChanged()
- (d) sensorUpdate()

Answer: (a) onSensorChanged()

Explanation: Called when sensor value changes

Session22

1. In FirebaseAuth, how do you sign in using a Google ID token?

- (a) signInWithEmail()
- (b) signInWithCredential(AuthCredential)
- (c) signInWithGoogle()
- (d) loginWithToken()

Answer: (b) signInWithCredential(AuthCredential)

2. Which class in Retrofit is responsible for defining the API interface?

- (a) RetrofitService
- (b) EndpointManager

(c) ApiService

(d) ApiClient

Answer: (c) ApiService

3. Which type of Firebase provider ID is returned after Google Sign-In?

(a) gmail.com

(b) auth.google.com

(c) firebase.google.com

(d) google.com

Answer: (d) google.com

4. Which annotation is used in Retrofit to send form-encoded data in a POST request?

(a) @Body

(b) @FormEncoded

(c) @FormUrlEncoded

(d) @Field

Answer: (c) @FormUrlEncoded

5. What does @SerializedName("key") do in Gson?

(a) Converts keys to camelCase

(b) Maps a JSON field to a Java variable

(c) Parses headers

(d) Encrypts data

Answer: (b) Maps a JSON field to a Java variable

6. What happens if you forget to add the Facebook App ID in strings.xml?

(a) User gets logged out

(b) Facebook login still works

(c) App crashes during Facebook login

(d) Login button doesn't appear

Answer: (c) App crashes during Facebook login

7. In Facebook Login SDK, what is the use of CallbackManager?

(a) Manage access tokens

(b) Track login attempts

(c) Handle login responses

(d) Bind Firebase

Answer: (c) Handle login responses

8. What role does the SHA-1 key play in social login integration with Firebase?

(a) It identifies your app to the provider

- (b) It encrypts data
- (c) It signs the user token
- (d) It hashes passwords

Answer: (a) It identifies your app to the provider

9. Which Firebase method is used to get user email after login?

- (a) fetchEmail()
- (b) getEmail()
- (c) userEmail()
- (d) getUserEmail()

Answer: (b) getEmail()

10. What will response.body() return in Retrofit if server sends HTTP 500 error?

- (a) Null
- (b) Valid JSON object
- (c) The raw body
- (d) Parsed object with error

Answer: (a) Null

11. In JSON parsing, which Gson method is used to convert JSON string to Java object?

- (a) fromJson()
- (b) parseJson()
- (c) toObject()
- (d) convertFromJson()

Answer: (a) fromJson()

12. Twitter Login with Firebase requires which dependency to be added?

- (a) firebase-oauth-login
- (b) firebase-auth only
- (c) firebase-twitter-login
- (d) firebase-auth and twitter-core

Answer: (d) firebase-auth and twitter-core

13. Which class represents the currently signed-in user on Google?

- (a) FirebaseAuth
- (b) GoogleSignInAccount
- (c) GoogleAccount
- (d) GoogleAuthClient

Answer: (b) GoogleSignInAccount

14. What is the typical file where Google Sign-In configuration is stored for Firebase integration?

- (a) google_auth.xml
- (b) firebase_config.xml
- (c) firebase-service.json
- (d) google-services.json

Answer: (d) google-services.json

15. What is the correct way to parse a nested JSON object using Gson?

- (a) Create matching nested POJO classes
- (b) Use JSONPath
- (c) Use JSONObject
- (d) Use @Expose

Answer: (a) Create matching nested POJO classes

16. What is the base class used by Retrofit to represent the HTTP response from a network call?

- (a) RetrofitResponse
- (b) Call<Response>
- (c) Response<T>
- (d) HttpResponse

Answer: (c) Response<T>

17. What does .addConverterFactory(GsonConverterFactory.create()) do in Retrofit?

- (a) Enables JSON serialization/deserialization
- (b) Allows multi-threading
- (c) Allows streaming JSON
- (d) Converts XML responses

Answer: (a) Enables JSON serialization/deserialization

18. What happens when an API returns non-2xx response in Retrofit?

- (a) App crashes
- (b) It triggers onFailure()
- (c) It throws IOException
- (d) It goes to onResponse() with response.isSuccessful() == false

Answer: (d) It goes to onResponse() with response.isSuccessful() == false

19. What does .signOut() do in Firebase Authentication?

- (a) Logs user out from all sessions
- (b) Signs out user locally
- (c) Deletes user account
- (d) Logs user out and clears Firebase project

Answer: (b) Signs out user locally

20. Which function is used to get the current signed-in user from FirebaseAuth?

- (a) `getSignedUser()`
- (b) `getCurrentUser()`
- (c) `getFirebaseSession()`
- (d) `isLoggedIn()`

Answer: (b) `getCurrentUser()`

21. What role does GsonConverterFactory play in Retrofit?

- (a) Handles UI binding
- (b) Handles database conversion
- (c) Maps Java objects to JSON
- (d) Handles network security

Answer: (c) Maps Java objects to JSON

22. How does Retrofit know which base URL to use for the API?

- (a) `@GET`
- (b) `Retrofit.Builder().baseUrl()`
- (c) `@BasePath`
- (d) API Interface

Answer: (b) `Retrofit.Builder().baseUrl()`

23. What is the use of `@Field` annotation in Retrofit?

- (a) Send JSON body
- (b) Used only with `@GET`
- (c) Send form data fields
- (d) Used for file uploads

Answer: (c) Send form data fields

24. Which Retrofit annotation allows sending dynamic URL parameters?

- (a) `@Path`
- (b) `@Header`
- (c) `@Query`
- (d) `@Param`

Answer: (a) `@Path`

25. What happens if you call `getCurrentUser()` after `signOut()`?

- (a) Returns null
- (b) Returns cached user
- (c) Throws error
- (d) Creates new user

Answer: (a) Returns null

26. What does GoogleSignIn.getClient() return?

- (a) FirebaseUser
- (b) Google ID token
- (c) GoogleSignInClient
- (d) GoogleSignInAccount

Answer: (c) GoogleSignInClient

27. What should be passed in signInWithCredential() for Google login?

- (a) Token string
- (b) AuthCredential
- (c) Email and password
- (d) OAuth2Key

Answer: (b) AuthCredential

28. What is required in Firebase Console to enable Facebook Login?

- (a) Facebook Graph API
- (b) SHA-256 only
- (c) Facebook App ID and Secret
- (d) Facebook XML config

Answer: (c) Facebook App ID and Secret

29. In Retrofit, what type of object is used for network callback?

- (a) Callback<T>
- (b) Future<T>
- (c) Observer<T>
- (d) Promise<T>

Answer: (a) Callback<T>

30. What is the role of Gson in JSON parsing with Retrofit?

- (a) UI Formatting
- (b) XML Decoding
- (c) Object Mapping
- (d) Authentication

Answer: (c) Object Mapping

31. What Retrofit method makes an API call asynchronously?

- (a) execute()
- (b) enqueue()
- (c) callNow()
- (d) sync()

Answer: (b) enqueue()

32. What's the correct way to create a Retrofit instance?

- (a) new Retrofit()
- (b) Retrofit.create()
- (c) Retrofit.Builder().build()
- (d) Retrofit.init()

Answer: (c) **Retrofit.Builder().build()**

33. Which HTTP method is used with @GET annotation in Retrofit?

- (a) POST
- (b) PUT
- (c) GET
- (d) DELETE

Answer: (c) **GET**

Explanation: @GET is a Retrofit annotation used to send HTTP GET requests to fetch data from a server.

34. What does @Query("user_id") in a Retrofit interface do?

- (a) Adds to the body of the request
- (b) Adds a query string like ?user_id=value
- (c) Defines a path parameter
- (d) Sends form-encoded data

Answer: (b) **Adds a query string like ?user_id=value**

Explanation: @Query appends the specified key-value pair as a query parameter in the URL.

35. Which method must be called on the Retrofit Call object to trigger the network request asynchronously?

- (a) run()
- (b) start()
- (c) enqueue()
- (d) init()

Answer: (c) **enqueue()**

Explanation: enqueue() triggers the API request on a background thread and handles the response via callback.

36. What is the purpose of the @Body annotation in Retrofit?

- (a) Sends a file
- (b) Sends form data
- (c) Sends a raw JSON object
- (d) Adds a query parameter

Answer: (c) **Sends a raw JSON object**

Explanation: @Body is used to send JSON or model data as the request body for POST/PUT requests.

37. Which Retrofit feature helps convert JSON response to a Java class?

- (a) RxJava
- (b) Interceptor
- (c) Converter Factory
- (d) Logger

Answer: (c) Converter Factory

Explanation: Retrofit uses Converter Factories like GsonConverterFactory to map JSON to Java objects.

38. What information must be added in GoogleSignInOptions.Builder for Google login?

- (a) Project name
- (b) Server API Key
- (c) RequestIdToken
- (d) Firebase Token

Answer: (c) RequestIdToken

Explanation: To authenticate with Firebase, you must request the ID token from Google using requestIdToken().

39. Which component is responsible for receiving Twitter access tokens in Firebase?

- (a) TwitterCore
- (b) TwitterConfig
- (c) FirebaseTwitterClient
- (d) TwitterAuthProvider

Answer: (d) TwitterAuthProvider

Explanation: TwitterAuthProvider is used to authenticate Twitter credentials with Firebase Authentication.

40. What is the main purpose of FirebaseAuth.getInstance()?

- (a) Creates a new Firebase project
- (b) Fetches database reference
- (c) Returns the default authentication instance
- (d) Signs in a user

Answer: (c) Returns the default authentication instance

Explanation: FirebaseAuth.getInstance() gives access to Firebase Authentication APIs for login/logout and user info.

41. Which exception is typically thrown when parsing a malformed JSON using Gson?

- (a) NullPointerException
- (b) JSONException
- (c) JsonSyntaxException
- (d) IllegalArgumentException

Answer: (c) JsonSyntaxException

Explanation: Gson throws JsonSyntaxException when JSON is invalid or does not match the target class structure.

42. What is the role of GoogleSignInClient in Google Login?

- (a) Checks if user exists in Firebase
- (b) Opens Google Auth Dialog
- (c) Parses ID token
- (d) Deletes Google Account

Answer: (b) Opens Google Auth Dialog

Explanation: GoogleSignInClient starts the intent that allows users to select a Google account to log in.

43. Which library is commonly used for converting complex JSON to Java objects?

- (a) Gson
- (b) Picasso
- (c) Glide
- (d) FirebaseUI

Answer: (a) Gson

Explanation: Gson is a widely-used library for converting JSON strings into Java objects and vice versa.

44. What is returned by response.body() when using Retrofit and the response is successful?

- (a) JSON string
- (b) Java object mapped by Gson
- (c) Retrofit Call object
- (d) null always

Answer: (b) Java object mapped by Gson

Explanation: response.body() returns a mapped Java object if the HTTP response is successful and a converter is used.

45. What should be included in AndroidManifest.xml when integrating Facebook Login SDK?

- (a) GoogleService.json
- (b) FacebookService.xml
- (c) Facebook App ID and meta-data
- (d) SHA key

Answer: (c) Facebook App ID and meta-data

Explanation: Facebook Login requires adding the App ID and metadata in the manifest for SDK to function properly.

46. What does Retrofit primarily help with in Android development?

- (a) Displaying UI
- (b) Accessing sensors
- (c) Making network requests
- (d) Storing preferences

Answer: (c) Making network requests

Explanation: Retrofit is a type-safe HTTP client for Android that simplifies network communication.

47. Which JSON format is valid?

- (a) {name: "John"}
- (b) {"name": John}
- (c) {"name": "John"}
- (d) {"name": "John"}

Answer: (d) {"name": "John"}

Explanation: JSON requires keys and string values to be enclosed in double quotes.

48. What is the default return type in a Retrofit API call interface?

- (a) Object
- (b) Call<T>
- (c) ResponseBody
- (d) String

Answer: (b) Call<T>

Explanation: Retrofit interfaces typically return Call<T> where T is the expected response type.

49. In JSON, what is the format for an array of strings?

- (a) {"name": ["Alice", "Bob"]}
- (b) {"name": "Alice, Bob"}
- (c) {"name": {"Alice", "Bob"}}
- (d) "name": [Alice, Bob]

Answer: (a) {"name": ["Alice", "Bob"]}

Explanation: A JSON array is enclosed in square brackets and elements are quoted if strings.

50. What is the minimum SDK requirement for using Google Sign-In?

- (a) 14
- (b) 15
- (c) 16
- (d) 19

Answer: (c) 16

Explanation: Google Sign-In APIs typically require API level 16 or higher for stable integration.

51. What does .create(ApiService.class) do in Retrofit?

- (a) Creates a new request
- (b) Converts JSON
- (c) Initializes the API interface
- (d) Closes network connection

Answer: (c) Initializes the API interface

Explanation: Retrofit uses .create() to generate implementation of the interface for making API calls.

52. What is the purpose of @POST annotation in Retrofit?

- (a) Defines URL path
- (b) Marks function to send a POST request
- (c) Declares return type
- (d) Used for authentication

Answer: (b) Marks function to send a POST request

Explanation: @POST tells Retrofit to use HTTP POST method for the annotated function.

53. Which social login requires google-services.json file in Android project?

- (a) Facebook Login
- (b) Twitter Login
- (c) Google Sign-In
- (d) Firebase Email/Password

Answer: (c) Google Sign-In

Explanation: Google Sign-In requires google-services.json for Firebase integration and configuration.

54. What method retrieves the ID of the currently signed-in Firebase user?

- (a) getUserId()
- (b) getUid()
- (c) getToken()
- (d) getIdToken()

Answer: (b) getUid()

Explanation: getUid() gives the unique ID assigned to the current Firebase user.

55. In Retrofit, what does @GET("users") represent?

- (a) JSON response type
- (b) Network timeout
- (c) URL endpoint
- (d) HTTP method type

Answer: (c) URL endpoint

Explanation: The string inside @GET annotation represents the relative URL path to the endpoint.

SESSION23

1. Which component must be extended to receive push notifications using Firebase?

- (a) JobService
- (b) FirebaseMessagingService
- (c) IntentService
- (d) BroadcastReceiver

Answer: (b) FirebaseMessagingService

Explanation: To handle FCM push messages, you must extend FirebaseMessagingService.

2. What is typically included in the payload of an FCM push notification?

- (a) Only title
- (b) Notification and Data objects
- (c) APK file
- (d) Firebase auth token

Answer: (b) Notification and Data objects

Explanation: FCM payload can include both notification (for UI) and data (for background logic) components.

3. Which design pattern encourages a *passive View* where the View delegates all actions to the Presenter?

- (a) MVC
- (b) MVVM
- (c) MVP
- (d) Singleton

Answer: (c) MVP

Explanation: MVP separates UI logic from business logic using a Presenter that updates the View.

4. In MVVM architecture, which component uses LiveData to notify UI changes?

- (a) Model
- (b) Activity
- (c) ViewModel
- (d) RecyclerView

Answer: (c) ViewModel

Explanation: ViewModel uses LiveData to automatically update the View when data changes.

5. Which of the following is NOT a valid benefit of using MVVM architecture?

- (a) Testable business logic
- (b) View and Model are tightly coupled
- (c) Use of data binding
- (d) Improved separation of concerns

Answer: (b) View and Model are tightly coupled

Explanation: MVVM avoids tight coupling between View and Model for better scalability.

6. Which file format is now recommended by Google Play for uploading apps?

- (a) .zip
- (b) .apk
- (c) .jar
- (d) .aab

Answer: (d) .aab

Explanation: Android App Bundle (.aab) is recommended for optimized app delivery on Google Play.

7. What must be done to an Android app before uploading to Google Play?

- (a) Clear logcat
- (b) Enable USB debugging
- (c) Generate a signed release build
- (d) Install on an emulator

Answer: (c) Generate a signed release build

Explanation: The app must be signed with a release key before publishing.

8. In MVP architecture, who directly updates the UI?

- (a) ViewModel
- (b) Model
- (c) View
- (d) Presenter

Answer: (d) Presenter

Explanation: The Presenter receives data from the Model and updates the View.

9. In MVVM, which Android component is commonly used for observing LiveData?

- (a) Fragment
- (b) ViewModelProvider
- (c) LifecycleOwner
- (d) Context

Answer: (c) LifecycleOwner

Explanation: LiveData observes LifecycleOwner components like Activity or Fragment.

10. Which method is used in FirebaseMessagingService to receive a new token for push notifications?

- (a) onTokenRefreshed()
- (b) onNewToken()
- (c) onMessageReceived()
- (d) getToken()

Answer: (b) onNewToken()

Explanation: onNewToken() is triggered when a new FCM token is generated.

11. What is the maximum size allowed for the data payload in an FCM message?

- (a) 1 KB
- (b) 4 KB
- (c) 2 KB
- (d) 5 KB

Answer: (b) 4 KB

Explanation: FCM supports up to 4 KB of data payload in downstream messages.

12. Which architecture pattern is ideal when the application requires **UI to auto-update** from model changes?

- (a) MVP
- (b) MVVM
- (c) MVC
- (d) Singleton

Answer: (b) MVVM

Explanation: MVVM allows Views to observe LiveData in ViewModel for auto UI updates.

13. Where do you declare push notification permissions in AndroidManifest.xml?

- (a) Inside <uses-sdk>
- (b) Under <activity>
- (c) Inside <uses-permission>
- (d) In ProGuard rules

Answer: (c) Inside <uses-permission>

Explanation: You declare necessary permissions like INTERNET and RECEIVE under <uses-permission>.

14. Which option is **NOT** part of the Google Play app publishing checklist?

- (a) Content rating form
- (b) Privacy policy URL
- (c) Firebase authentication setup
- (d) App description and screenshots

Answer: (c) Firebase authentication setup

Explanation: Firebase Auth is not mandatory for Play Store submission unless used in the app.

15. What is the minimum API level required to use Android App Bundles (.aab)?

- (a) 16
- (b) 21
- (c) 24
- (d) 19

Answer: (b) 21

Explanation: Android App Bundle is supported from Android 5.0 (API level 21) and above.

31. What method must be overridden to receive push notifications using FirebaseMessagingService?

- (a) onStart()
- (b) **onMessageReceived(RemoteMessage remoteMessage)**
- (c) onNotificationPosted()
- (d) sendPushNotification()

Answer: (b)

Explanation: The `onMessageReceived()` method is used to handle incoming FCM messages in `FirebaseMessagingService`.

32. In MVVM, who should trigger the API call to fetch data from the server?

- (a) View
- (b) ViewModel**
- (c) DataBindingUtil
- (d) Repository

Answer: (b)

Explanation: The ViewModel initiates data operations, keeping UI code separate and testable.

33. Which architectural component in MVVM typically communicates with REST APIs or local databases?

- (a) Presenter
- (b) LiveData
- (c) View
- (d) Repository**

Answer: (d)

Explanation: The Repository handles data operations and acts as a single source of truth for ViewModel.

34. What Android class is used to schedule notification delivery even when the app is not running?

- (a) HandlerThread
- (b) WorkManager**
- (c) BroadcastReceiver
- (d) IntentService

Answer: (b)

Explanation: WorkManager is used to run background tasks reliably, including push notification scheduling.

35. In MVP, what should the View not do?

- (a) Display UI elements
- (b) Receive user input
- (c) Modify the data model**
- (d) Show error messages

Answer: (c)

Explanation: The View only interacts with the Presenter. It shouldn't directly change data.

36. When publishing to Play Store, what file contains versioning and signing config?

- (a) `AndroidManifest.xml`
- (b) `strings.xml`
- (c) `build.gradle`**

(d) MainActivity.java

Answer: (c)

Explanation: build.gradle includes app version, signing keys, and build configurations.

37. Which of the following is a major drawback of tightly coupling View and Model?

- (a) Easier to test
- (b) Fewer files to manage
- (c) Difficult to maintain and scale**
- (d) Better performance

Answer: (c)

Explanation: Tight coupling leads to poor modularity and maintainability.

38. What is the major difference between MVP and MVVM?

- (a) MVP uses ViewModel
- (b) MVVM uses data-binding for automatic UI updates**
- (c) MVP uses Repository
- (d) MVVM has no separation of concerns

Answer: (b)

Explanation: MVVM supports data binding to reduce boilerplate UI code; MVP requires manual updates.

39. Before uploading an app, which file should be signed using your keystore?

- (a) AndroidManifest.xml
- (b) APK or AAB file**
- (c) Gradle.properties
- (d) .java files

Answer: (b)

Explanation: APK/AAB files must be digitally signed before release to ensure authenticity and integrity.

40. What is the function of google-services.json in Firebase integration?

- (a) Handle login UI
- (b) Define app permissions
- (c) Configure Firebase project credentials**
- (d) Manage activity lifecycle

Answer: (c)

Explanation: This file connects your app to the Firebase project, storing app-specific settings.

41. In MVVM, which of the following allows separation of UI logic from business logic?

- (a) View
- (b) ViewModel**
- (c) Activity

(d) Handler

Answer: (b)

Explanation: ViewModel stores UI-related data separately from the View, supporting testability and lifecycle awareness.

42. What is one primary benefit of using architecture patterns like MVVM?

(a) Code duplication

(b) Separation of concerns and modularity

(c) Poor testability

(d) Tightly coupled components

Answer: (b)

Explanation: MVVM promotes modular code that's easier to test, maintain, and scale.

43. If an app uses Data Binding and LiveData, what is automatically updated when LiveData changes?

(a) Activity

(b) Bound UI element

(c) Repository

(d) Manifest

Answer: (b)

Explanation: LiveData automatically notifies bound UI elements when its value changes, no need to manually update.

44. What tool is required to upload an app bundle to Google Play Console?

(a) Android NDK

(b) AVD Manager

(c) Google Play Console

(d) Firebase Emulator

Answer: (c)

Explanation: The Play Console is used to manage, upload, and monitor Android apps for publishing.

45. Which of the following is a valid step in the app publishing process?

(a) Upload MainActivity.java

(b) Push to Firebase

(c) Create app listing with description and screenshots

(d) Zip entire project and upload

Answer: (c)

Explanation: A valid Play Store listing includes app name, description, screenshots, and category.