Session 14: Saving Data & Working with System Permissions

Assignment - 1

1. **What is the difference between Internal Storage & External Storage?**

Ans -

|  |  |
| --- | --- |
| Internal Storage | External Storage |
| 1. Use internal device storage for private data. 2. Store private data on the device memory. 3. It's always available. 4. Files saved here are accessible by only your app. 5. When the user uninstalls your app, the system removes all your app's files from internal storage. 6. Internal storage is best when you want to be sure that neither the user nor other apps can access your files. 7. The Internal Storage is reserved for your Operating System. 8. Memory is less. 9. No permissions are required to save files on the internal storage. Your application always has permission to read and write files in its internal storage directory. | 1. Use external storage for large data sets that are not private. 2. Store public data on the shared external storage. 3. It's not always available, because the user can mount the external storage as USB storage and in some cases remove it from the device. 4. It's world-readable, so files saved here may be read outside of your control. 5. When the user uninstalls your app, the system removes your app's files from here only if you save them in the directory from [getExternalFilesDir()](https://developer.android.com/reference/android/content/Context.html" \l "getExternalFilesDir(java.lang.String)). 6. External storage is the best place for files that don't require access restrictions and for files that you want to share with other apps or allow the user to access with a computer. 7. Before you do any work with the external storage, you should always call [getExternalStorageState()](https://developer.android.com/reference/android/os/Environment.html" \l "getExternalStorageState()) to check whether the media is available. The media might be mounted to a computer, missing, read-only, or in some other state. 8. More memory. |

1. **For how long the data resides in the cache?**

Ans. –

1. Cached files on your Android phone or tablet can exist for the apps which you access or use, and it will exist for the websites you visit using your smartphone. All apps, whether they are system apps or third party apps will have cached data. Since cached data is automatically created and it does not include any important data, [wiping or clearing the cache](https://www.technobezz.com/how-to-wipe-cache-partition-on-htc-one-m9/)for an app or a device is harmless.
2. While cached data is helpful for computers, smartphones, and tablets; it can take up space on your device and it may, over a period of time, get corrupt. While the cached data on your device can be helpful for the websites you visit regularly, it can be considered a waste for those websites which you will be visiting just once.
3. Android devices manage app data very efficiently, and for most of the users, you will not have to worry about managing the cached data manually. However, when the phone begins to face issues and you are trying to fix a problem by following certain troubleshooting steps, then you may come across a step which may ask you to clear cache.
4. To preserve file space and maintain your app's performance, it's important that you carefully manage your cache files and remove those that aren't needed anymore throughout your app's lifecycle.
5. Clearing the files directory will also clear the cache directory.
6. Cache will be deleteted if user uninstalls the application.
7. File cacheDir = context.getCacheDir();//gets a reference to the cache directory
8. \* long size = getDirSize(cacheDir);//we can determine the space available in the cahce directory
9. \* getCacheDir()gives reference to internal cache.
10. \* getExternalCacheDir() for external storage specific to app path.
11. **What are the critical Permissions and Normal Permissions? What are the examples of each?**

Ans-

System permissions are divided into several protection levels. The two most important protection levels to know about are *normal* and critical permissions:

1. *Normal* permissions cover areas where your app needs to access data or resources outside the app's sandbox, but where there's very little risk to the user's privacy or the operation of other apps. For example, permission to set the time zone is a normal permission. If an app declares that it needs a normal permission, the system automatically grants the permission to the app. For a full listing of the current normal permissions,
2. Many permissions are designated as [PROTECTION\_NORMAL](https://developer.android.com/reference/android/content/pm/PermissionInfo.html#PROTECTION_NORMAL), which indicates that there's no great risk to the user's privacy or security in letting apps have those permissions. For example, users would reasonably want to know whether an app can read their contact information, so users have to grant this permission explicitly. By contrast, there's no great risk in allowing an app to vibrate the device, so that permission is designated as *normal.*
3. If an app declares in its manifest that it needs a normal permission, the system automatically grants the app that permission at install time. The system does not prompt the user to grant normal permissions, and users cannot revoke these permissions.
4. As of API level 23, the following permissions are classified as [PROTECTION\_NORMAL](https://developer.android.com/reference/android/content/pm/PermissionInfo.html#PROTECTION_NORMAL):

[ACCESS\_LOCATION\_EXTRA\_COMMANDS](https://developer.android.com/reference/android/Manifest.permission.html#ACCESS_LOCATION_EXTRA_COMMANDS)

[ACCESS\_NETWORK\_STATE](https://developer.android.com/reference/android/Manifest.permission.html#ACCESS_NETWORK_STATE)

[ACCESS\_NOTIFICATION\_POLICY](https://developer.android.com/reference/android/Manifest.permission.html#ACCESS_NOTIFICATION_POLICY)

[ACCESS\_WIFI\_STATE](https://developer.android.com/reference/android/Manifest.permission.html#ACCESS_WIFI_STATE)

[BLUETOOTH](https://developer.android.com/reference/android/Manifest.permission.html#BLUETOOTH)

[BLUETOOTH\_ADMIN](https://developer.android.com/reference/android/Manifest.permission.html#BLUETOOTH_ADMIN)

[BROADCAST\_STICKY](https://developer.android.com/reference/android/Manifest.permission.html#BROADCAST_STICKY)

[CHANGE\_NETWORK\_STATE](https://developer.android.com/reference/android/Manifest.permission.html#CHANGE_NETWORK_STATE)

[CHANGE\_WIFI\_MULTICAST\_STATE](https://developer.android.com/reference/android/Manifest.permission.html#CHANGE_WIFI_MULTICAST_STATE)

[CHANGE\_WIFI\_STATE](https://developer.android.com/reference/android/Manifest.permission.html#CHANGE_WIFI_STATE)

[DISABLE\_KEYGUARD](https://developer.android.com/reference/android/Manifest.permission.html#DISABLE_KEYGUARD)

[EXPAND\_STATUS\_BAR](https://developer.android.com/reference/android/Manifest.permission.html#EXPAND_STATUS_BAR)

[GET\_PACKAGE\_SIZE](https://developer.android.com/reference/android/Manifest.permission.html#GET_PACKAGE_SIZE)

[INSTALL\_SHORTCUT](https://developer.android.com/reference/android/Manifest.permission.html#INSTALL_SHORTCUT)

[INTERNET](https://developer.android.com/reference/android/Manifest.permission.html#INTERNET)

[KILL\_BACKGROUND\_PROCESSES](https://developer.android.com/reference/android/Manifest.permission.html#KILL_BACKGROUND_PROCESSES)

[MODIFY\_AUDIO\_SETTINGS](https://developer.android.com/reference/android/Manifest.permission.html#MODIFY_AUDIO_SETTINGS)

[NFC](https://developer.android.com/reference/android/Manifest.permission.html#NFC)

[READ\_SYNC\_SETTINGS](https://developer.android.com/reference/android/Manifest.permission.html#READ_SYNC_SETTINGS)

[READ\_SYNC\_STATS](https://developer.android.com/reference/android/Manifest.permission.html#READ_SYNC_STATS)

[RECEIVE\_BOOT\_COMPLETED](https://developer.android.com/reference/android/Manifest.permission.html#RECEIVE_BOOT_COMPLETED)

[REORDER\_TASKS](https://developer.android.com/reference/android/Manifest.permission.html#REORDER_TASKS)

[REQUEST\_IGNORE\_BATTERY\_OPTIMIZATIONS](https://developer.android.com/reference/android/Manifest.permission.html#REQUEST_IGNORE_BATTERY_OPTIMIZATIONS)

[REQUEST\_INSTALL\_PACKAGES](https://developer.android.com/reference/android/Manifest.permission.html#REQUEST_INSTALL_PACKAGES)

[SET\_ALARM](https://developer.android.com/reference/android/Manifest.permission.html#SET_ALARM)

[SET\_TIME\_ZONE](https://developer.android.com/reference/android/Manifest.permission.html#SET_TIME_ZONE)

[SET\_WALLPAPER](https://developer.android.com/reference/android/Manifest.permission.html#SET_WALLPAPER)

[SET\_WALLPAPER\_HINTS](https://developer.android.com/reference/android/Manifest.permission.html#SET_WALLPAPER_HINTS)

[TRANSMIT\_IR](https://developer.android.com/reference/android/Manifest.permission.html#TRANSMIT_IR)

[UNINSTALL\_SHORTCUT](https://developer.android.com/reference/android/Manifest.permission.html#UNINSTALL_SHORTCUT)

[USE\_FINGERPRINT](https://developer.android.com/reference/android/Manifest.permission.html#USE_FINGERPRINT)

[VIBRATE](https://developer.android.com/reference/android/Manifest.permission.html#VIBRATE)

[WAKE\_LOCK](https://developer.android.com/reference/android/Manifest.permission.html#WAKE_LOCK)

[WRITE\_SYNC\_SETTINGS](https://developer.android.com/reference/android/Manifest.permission.html#WRITE_SYNC_SETTINGS)

1. *Dangerous permissions* cover areas where the app wants data or resources that involve the user's private information, or could potentially affect the user's stored data or the operation of other apps. For example, the ability to read the user's contacts is a dangerous permission. If an app declares that it needs a dangerous permission, the user has to explicitly grant the permission to the app.
   1. All dangerous Android system permissions belong to permission groups. If the device is running Android 6.0 (API level 23) and the app's targetSdkVersion is 23 or higher, the following system behavior applies when your app requests a dangerous permission:
   2. If an app requests a dangerous permission listed in its manifest, and the app does not currently have any permissions in the permission group, the system shows a dialog box to the user describing the permission group that the app wants access to. The dialog box does not describe the specific permission within that group. For example, if an app requests the READ\_CONTACTS permission, the system dialog box just says the app needs access to the device's contacts. If the user grants approval, the system gives the app just the permission it requested.
   3. If an app requests a dangerous permission listed in its manifest, and the app already has another dangerous permission in the same permission group, the system immediately grants the permission without any interaction with the user. For example, if an app had previously requested and been granted the READ\_CONTACTS permission, and it then requests WRITE\_CONTACTS, the system immediately grants that permission.

## Dangerous permissions:

* 1. \* READ\_CALENDAR
  2. \* WRITE\_CALENDAR
  3. \* CAMERA
  4. \* READ\_CONTACTS
  5. \* WRITE\_CONTACTS
  6. \* GET\_ACCOUNTS
  7. \* ACCESS\_FINE\_LOCATION
  8. \* ACCESS\_COARSE\_LOCATION
  9. \* RECORD\_AUDIO
  10. \* READ\_PHONE\_STATE
  11. \* CALL\_PHONE
  12. \* READ\_CALL\_LOG
  13. \* WRITE\_CALL\_LOG
  14. \* ADD\_VOICEMAIL
  15. \* USE\_SIP
  16. \* PROCESS\_OUTGOING\_CALLS
  17. \* BODY\_SENSORS
  18. \* SEND\_SMS
  19. \* RECEIVE\_SMS
  20. \* READ\_SMS
  21. \* RECEIVE\_WAP\_PUSH
  22. \* RECEIVE\_MMS
  23. \* READ\_EXTERNAL\_STORAGE
  24. \* WRITE\_EXTERNAL\_STORAGE