**Applications of Threading in real life**

Here are some examples of how threading is used in various applications:

* Video and Audio Streaming: One thread may be used to download the media file from a server, while another thread decodes and plays the media on the user's device.
* Gaming: By using multiple threads, game developers can ensure that the game remains responsive to user input, even when dealing with multiple players and complex game logic.
* Web Browsing: By using multiple threads, browsers can load multiple resources simultaneously, making the browsing experience faster and more responsive.
* Data Processing: Each thread can perform a separate part of the calculation, allowing the application to complete the processing more quickly.
* Mobile and Embedded Devices: By using multiple threads, developers can make their applications more responsive and efficient, and can take advantage of multiple cores found in modern mobile and embedded devices.
* Digital Watch: Display Updates, in a digital watch, the display needs to be updated constantly to show the current time. By using multi-threading, the display update can be performed in a separate thread while the main thread handles user input and other tasks. This ensures that the watch remains responsive to user input, even while the display is being updated. Multi-threading is also used for sensor data processing. Many digital watches have sensors for measuring heart rate, steps, etc. These sensor data need to be processed in real-time to provide accurate readings. By using separate threads for sensor data processing, the main thread can continue to handle user interactions and other tasks without being blocked, resulting in a smoother user experience.
* Smart TV: Multi-threading is used for Video Playback. Smart TVs often need to handle video playback, which requires decoding and rendering of video frames. By using multi-threading, the video decoding and rendering can be performed in separate threads, allowing for smoother playback, and preventing the video from freezing or stuttering. Multi-threading is also used for background tasks. Smart TVs often have multiple background tasks running, such as downloading updates, syncing data, etc. By using threads, these background tasks can be performed in parallel without blocking the main thread that handles user input and UI updates, ensuring smooth performance and responsiveness of the smart TV.