Example 6:

Implement an application that uses Multi-Threading.

Multi-Threading:

Multi-Threading in Android is a unique feature through which more than one threads execute together without hindering the execution of other threads. Multi-Threading in Android is not different from conventional multi-Threading. A class can be thought of as a process having its method as it's sub-processes or threads. All these methods can run concurrently by using feature of Multi-Threading. In android, multi-Threading can be achieved through the use of many in-built classes. Out of them, Handler class is most commonly used.

Handler class in Android:

Handler class come from the Package android.os.Handler package and is most commonly used for multi-threading in android. Handler class provide sending and receiving feature for messages between different threads and handle the thread execution which is associated with that instance of Handler class.

In android class, every thread is associated with an instance of Handler class and it allows the thread to run along with other threads and communicate with them through messages.

Instantiating Handler class:

There are following two ways in which Handler class is usually instantiated for supporting multi-threading:

- Through default constructor.

  Handler handlerObject = new Handler();
- Through Parameterized constructor

  Handler handleObject = new Handler(Runnable runnableObject, Handler.Callback callbackObject);

Methods of Handler class for Multi-Threading:

- **Public final Boolean post(Runnable runnableObject){ return booleanValue; }**

This Method attach a runnable instance with it's associated thread and the body of that runnable instance will execute every time the thread gets executed.

- **Public final Boolean postAtTime((Runnable runnableObject, long timeinMillisecondObject){ return booleanValue; }**

This Method attach a runnable instance with it’s associated thread and the body of that runnable instance will execute every time the thread gets executed at a time specified by the second argument.

- **Public final Boolean postDelayed((Runnable runnableObject, long timeinMillisecondObject){ return booleanValue; }**

This Method attach a runnable instance with its associated thread and the body of that runnable instance will execute every time the thread gets executed after a time specified by the second argument.

Runnable Interface:

Runnable interface is used in multi-threading to be called in a loop when the thread starts.

It is a type of thread that executes the statement in its body or calls other methods for a specified or infinite number of times.

This runable interface is used by the Handler class to execute the multi-threading, i.e., to execute one or more thread in specified time.

Runnable is an interface which is implemented by the class desired to support multithreading and that class must implements it’s abstract method public void run().

run() method is the core of multithreading as it includes the statement or calls to other methods that the thread needs to be made for multithreading.

```
class ClassName implements Runnable
{
            @override
            Public void run()
            {
                        Body of method
            }
}
```

Runnable interface can also be used by using adapter class as explained below:

```
Runnable runnableObject =new Runnable()
{
        @override
        Public void run()
        {

        };
}
```