DAEMON THREAD:-

----------------

\* Daemon thread is considered as helping thread because it will helps the threds which is running in the application.

\* Daemon thread always executes in the background.

\* This thread will run in background from the begging to ending of the program.

example for daemon thread is "garbage collector","auto-spell check","auto-compiler","signal dispatcher"....etc:

inbuilt methods w.r.t Daemon thread.

-------------------------------------

1. public boolean isDaemon()

--> It will return the boolean result as true or false which indicates wheter the thread is daemon or non-daemon thread.

2. public void setDaemon(boolean b)

--> by using this we can make the non-daemon thread as daemon and vice-versa.

EXAMPLE:

---------

OUTPUT:

---------

EXAMPLE:

---------

OUTPUT:

---------

NOTE: Daemon thread ends when the all the non-daemon thread completes its execution.

but, even though the daemon threads are executing in the background jvm will destroy them immediately after the completion of non-daemon thread.

INTER-THREAD COMMUNICATIONS:

--------------------------------

when ever multi-threads are executing simultaneously in the multithreaded application there may be chance of thread dependency.

when ever thread dependency exist then there must a communication b/w these threads.

The communication b/w the threads of the same application is called as "Inter-Thread Communication".

To achive Inter thread communication we will make use of 3 methods:

1. wait()

2. notify()

3. notifyall()

Note: All the 3 methods are present in "object class" and not in Thread class.

wait() method:-

----------------

wait method is alternative for join method. When wait() is called the current thread goes to waiting state untill the other thread notify it.

The Thread which made another thread to wait will execute first then after executing it will notify the thread which is in wait state.

wait() is overloaded method in thread class

--> wait()

--> wait(long ms)

--> wait(long ms, int ns)

notify() method:-

-----------------

notify method is called by the executing thread to the waiting thread to indicate that thread to come out of waiting state.

NOTE: wait(), notify(), notifyall() all these method should be written inside synchronized block.

difference b/w notify() and notifyall() ?

--> notify() method will notify only one thread at a time. notifyall() method will notify all the thread at a time.

EXAMPLE:

--------

OUTPUT:

--------

PRODUCER AND CONSUMER PROBLEM TO EXPLAIN INTER THREAD COMMUNICATION:

---------------------------------------------------------------------

SCENARIO:

---------

In the producer and consumer problem the time taken for the production and for consumption would vary.

As long as the producer produce the item consumer can not cosume it.

In order to bring the communication b/w producer and consumer we need to use wait() and notify() method.

REFER DIA:

EXAMPLE:

--------

OUTPUT:

-------

DEAD LOCK and STARVATION:-

---------------------------

If thread goes for waiting state for infinite time then it is called as DEAD LOCK.

If thread goes for waiting state for long period but after long time if its come to normal state then is called as STARVATION.

NOTE: generally thread will go to starvation when ever all the other threads priority is higher than its prioirty.

EXAMPLE:

--------

OUTPUT:

-------