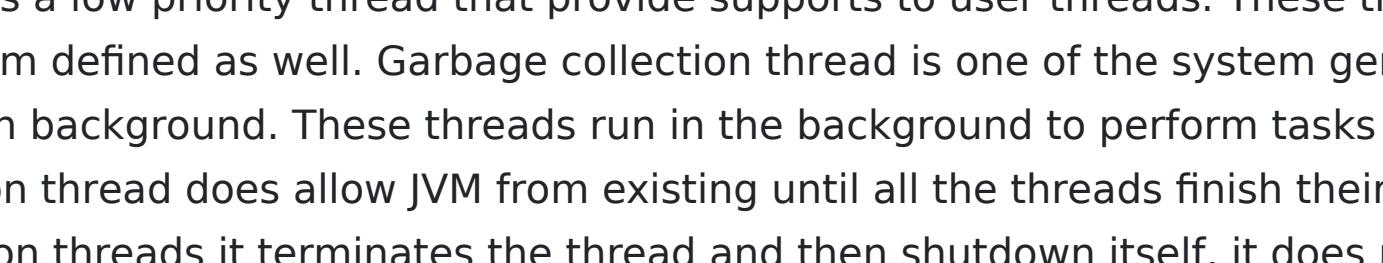
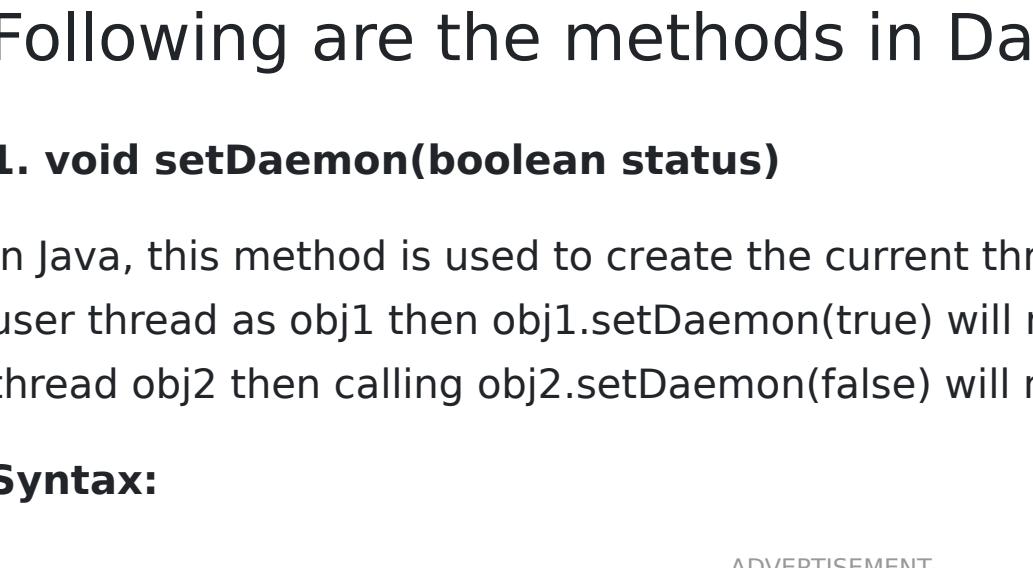


## Java Daemon Thread

ADVERTISEMENT



Daemon threads is a low priority thread that provide supports to user threads. These threads can be user defined and system defined as well. Garbage collection thread is one of the system generated daemon thread that runs in background. These threads run in the background to perform tasks such as garbage collection. Daemon thread does allow JVM from existing until all the threads finish their execution. When a JVM finds daemon threads it terminates the thread and then shutdown itself, it does not care Daemon thread whether it is running or not.



Normal Thread

Daemon Thread

Following are the methods in Daemon Thread

### 1. void setDaemon(boolean status)

In Java, this method is used to create the current thread as a daemon thread or user thread. If there is a user thread as obj1 then obj1.setDaemon(true) will make it a Daemon thread and if there is a Daemon thread obj2 then calling obj2.setDaemon(false) will make it a user thread.

#### Syntax:

ADVERTISEMENT

```
public final void setDaemon(boolean on)
```

### 2. boolean isDaemon()

In Java, this method is used to check whether the current thread is a daemon or not. It returns true if the thread is Daemon otherwise it returns false.

#### Syntax:

```
public final boolean isDaemon()
```

#### Example:

Lets create an example to create daemon and user threads. To create daemon thread setdaemon() method is used. It takes boolean value either true or false.

```
public class DaemonDemo1 extends Thread
{
    public DaemonDemo1(String name)
    {
        super(name);
    }
    public void run()
    {
        if(Thread.currentThread().isDaemon())
        {
            System.out.println(getName() + " is Daemon thread");
        }
        else
        {
            System.out.println(getName() + " is User thread");
        }
    }
    public static void main(String[] args)
    {
    }
}
```

## OUTPUT:

```
C:\Windows\System32\cmd.exe
D:\Studytonight>javac DaemonDemo1.java
D:\Studytonight>java DaemonDemo1
D1 is Daemon thread
D3 is Daemon thread
D2 is User thread
```

#### Example : Daemon thread Priority

Since daemon threads are low level threads then lets check the priority of these threads. The priority we are getting is set by the JVM.

```
public class DaemonDemo1 extends Thread
{
    public DaemonDemo1(String name)
    {
        super(name);
    }
    public void run()
    {
        if(Thread.currentThread().isDaemon())
        {
            System.out.println(getName() + " is Daemon thread");
        }
        else
        {
            System.out.println(getName() + " is User thread");
        }
        System.out.println(getName()+" priority "+Thread.currentThread().getPriority());
    }
}
```

## OUTPUT:

```
D1 is Daemon thread
D1 priority 5
D2 is User thread
D3 is Daemon thread
D2 priority 5
D3 priority 5

D3 priority 5
at java.base/java.lang.Thread.setDaemon(Thread.java:1410)
at myjavaproject.DaemonDemo1.main(DaemonDemo1.java:32)
```

← Prev

Next →

## Java MCQ Tests

Prepare for Java Interview in TCS, Infosys, etc. companies.

Explore

## Java Programs

Java programs with code and output for practice.

Explore

## Spring Framework

Learn the most widely used Java framework in the World.

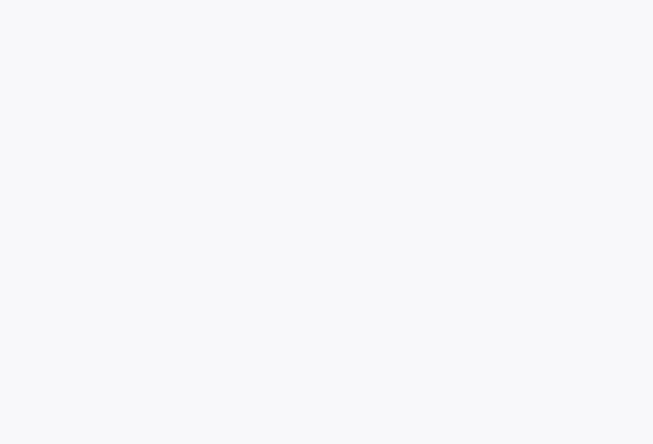
Explore

ADVERTISEMENT

**Get FREE protection for your online life.**

DOWNLOAD AVAST ONE  
Avast One

ADVERTISEMENT



IRG Cayman Open &gt;

No compatible source was found for this media.

Hello X Trends

## BASICS OF JAVA ▾

## OOPS CONCEPTS ▾

## STRING HANDLING ▾

## EXCEPTION HANDLING ▾

## JAVA MULTITHREADING ▾

Introduction to Multithreading

Thread Class

Creating a thread

Joining threads

Sleeping Thread in Java

Naming Thread in Java

Thread Priority in Java

### Daumen Thread in Java

Synchronization

Interthread Communication

Thread group

## ADVANCED TOPICS ▾

### COLLECTION FRAMEWORK

## JAVA GUI ▾

## REFLECTION API ▾

## RMI APPLICATION ▾

## INNER CLASS ▾

## WRAPPER CLASS ▾

## FILE HANDLING ▾

## LIST ▾

## SET ▾

## MAP ▾

## QUEUE & DEQUE ▾

## JDBC ▾

## LAYOUT MANAGERS ▾

ADVERTISEMENT

D X

Advertisement

Advertisement