# 

[Multithreading:](#_az9vpmupaob)

[Thread Creation](#_xd561jivesg9)

[Thread Join](#_six1pca6120d)

[Synchronised](#_d7aweh617rbb)

[Interthread communication](#_8k6dr6fb9mue)

[Program 1](#_2lga51t3qls9)

[Program 2](#_iuv6kp2mvrb0)

# 

# Multithreading:

A thread is nothing but an independent path followed while executing a program. Every Java program has one thread by default.Yes, the thread invoked by calling the main() method. It is called main thread.

The benefit of multithreading in Java is the fact that one thread doesn’t have to depend on another for its execution. They can each independently coexist without affecting each other’s result. So say one thread pauses, the others could still keep on getting executed.

## Thread Creation

Thread can be created in two ways:

1.extends thread class

2. Implements runnable interface

## Thread Join

To make a thread to wait and to start the new thread is where join is used.

For eg:

t1.start();

t1.join(); // after it dies t2 is started.

t2.start();

t2.join();

## Synchronised

In a program to write to a file then read, the reading thread should wait for the writing thread to finish before beginning to read. To enforce such synchronicity, Java makes use of a synchronization process known as the monitor.

## Interthread communication

wait()--to make a thread to go into waiting start.

notify()--to notify a thread that it should be started.

notifyAll()--to notify and initiate all the thread.

# Program 1

Given three arrays

Array 1 = [2,4,6,8,10,12];

Array 2 = [12,14,16,18,20,22];

Array 3 = [22,24,26,28,30,32];

Create three threads thread1,thread2 and thread3 that should access individual arrays.

Output : 2 12 22 4 14 24 6 16 26 8 18 28 10 20 30 12 22 32

Use synchronise to solve this.Make first thread to run and display the first element in the first array allow the thread to wait and now 2nd thread runs and displays the first element in the second array and similarly the third thread.Then notify the first thread.

# Program 2

Given two arrays of same length

Array 1 = [1,2,3,4,5];

Array 2 = [2,3,4,5,6];

Square the elements in the first array.Cube all the elements in the second array.

Add the elements of first array with elements of second array.

Print the resultant array.Use appropriate multithreading techniques.

Create three threads.one for square, second thread for cubing and third thread to sum and display.

Thread one and two can be run separate thus

t1.start();

t2.start();

But before the third thread starts, thread one and two must complete squaring and cubing.Thus make thread one and two to die and start third thread.

t1.join();

t2.join();

t3.start();