MULTITHREADING

[CREATING THREADS:](#_odpr4qz9fetv)

[JOINING THREADS:](#_yo2hcn2djox5)

[THREAD SYNCHRONIZATION:](#_rvnhen9nf4gz)

[INTER-THREAD COMMUNICATION:](#_vi5sod8m4obt)

[QUESTION1:](#_m3056zo2ir9u)

[Solved Question 1:](#_64eq3kuk0dx)

[QUESTION2:](#_bmuiks4kpdmx)

[Solved Question 2:](#_vzi88vvt6o3w)

### CREATING THREADS:

There are two ways in which thread can be created.

Extending Thread class

Implementing runnable interface

### JOINING THREADS:

Use to execute one thread after another thread has completed its execution.

t1.join();

### THREAD SYNCHRONIZATION:

When multiple threads try to access same resource, synchronization is used.

### INTER-THREAD COMMUNICATION:

**wait()**  tells the calling thread to give up the monitor and go to sleep until some other thread enters the same monitor and calls notify().

**notify()** wakes up the first thread that called wait() on the same object.

**notifyAll()** wakes up all the threads that called wait() on the same object. The highest priority thread will run first.

## QUESTION1:

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

#### Solved Question 1:

<https://github.com/sruthiviswanathan/Multithreading/blob/master/src/com/zilker/multithreading/Multiprogram.java>

* Declare and initialize 3 arrays.
* Create 3 threads and execute t1.thread(),t2.thread(),t3.thread() sequentially.
* run() method has the condition that prints each array sequentially.(i.e) if the first array is executed then the condition for second array is enabled and the third array is diabled.similarly do for the other threads.

## QUESTION2:

<https://github.com/sruthiviswanathan/Multithreading/blob/master/src/com/zilker/multithreading/Multithreadsum.java>

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.

#### Solved Question 2:

* Declare and initialize 2 arrays.
* Create 3 threads
* Have 1st thread for squaring and store it in an array,
* Have 2nd thread for performing cube and store it separately in another array.
* Make the third thread wait till 1st and 2nd thread completes execution.
* Now use 3rd thread to sum both and print.