

Newer : Vishal Sharma

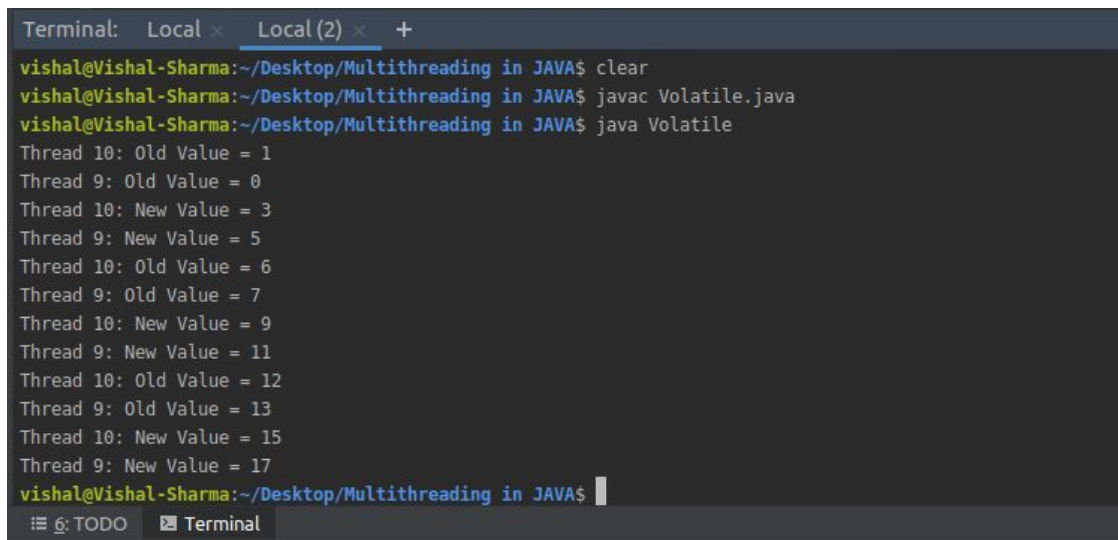
Newer ID : 4171

Email : vishal.sharma@tothenew.com

Exercise : Multithreading in JAVA

1. Write a program to demonstrate the use of volatile keyword.

Answer: Filename- **Volatile.java**



```
Terminal: Local x Local (2) x +
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ clear
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ javac Volatile.java
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ java Volatile
Thread 10: Old Value = 1
Thread 9: Old Value = 0
Thread 10: New Value = 3
Thread 9: New Value = 5
Thread 10: Old Value = 6
Thread 9: Old Value = 7
Thread 10: New Value = 9
Thread 9: New Value = 11
Thread 10: Old Value = 12
Thread 9: Old Value = 13
Thread 10: New Value = 15
Thread 9: New Value = 17
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$
```

2. Write a program to create a thread using Thread class and Runnable interface each.

Answer: Filename- **RunnableUse.java**

```
File Edit View Search Terminal Help
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ javac RunnableUse.java
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ java RunnableUse
Inside thread class in anonymous 0
Inside thread class in anonymous 1
Inside thread class in Runnable 0
Inside thread class in anonymous 2
Inside thread class in Runnable 1
Inside thread class in anonymous 3
Inside thread class in Runnable 2
Inside thread class in anonymous 4
Inside thread class in anonymous 5
Inside thread class in Runnable 3
Inside thread class in anonymous 6
Inside thread class in Runnable 4
Inside thread class in anonymous 7
Inside thread class in Runnable 5
Inside thread class in anonymous 8
Inside thread class in Runnable 6
Inside thread class in Runnable 7
Inside thread class in Runnable 8
Inside thread class in Runnable 9
Inside thread class in anonymous 9
Inside Thread Class 0
Inside Thread Class 1
Inside Thread Class 2
Inside Thread Class 3
Inside Thread Class 4
Inside Thread Class 5
Inside Thread Class 6
Inside Thread Class 7
Inside Thread Class 8
Inside Thread Class 9
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$
```

3. Write a program using synchronization block and synchronization method

Answer: Filename- **Sync.java**

```
Terminal: Local x Local (2) x +
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ javac Sync.java
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ java Sync
Inside thread class using sync method0
Inside thread class using sync method0
Inside thread class using sync method1
Inside thread class using sync method1
Inside thread class using sync method2
Inside thread class using sync method2
Inside thread class using sync method3
Inside thread class using sync method3
Inside thread class using sync method4
Inside thread class using sync method4
Inside thread class using sync method5
Inside thread class using sync method5
Inside thread class using sync method6
Inside thread class using sync method6
Inside thread class using sync method7
Inside thread class using sync method7
Inside thread class using sync method8
Inside thread class using sync method8
Inside thread class using sync method9
Inside thread class using sync method9
Count = 20
Count = 20
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$
```

4. Write a program to create a Thread pool of 2 threads where one Thread will print even numbers and other will print odd numbers.

Answer: Filename- **OddEven.java**

```
Terminal: Local x Local (2) x +
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ javac OddEven.java
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ java OddEven
Odd Number 1
Odd Number 3
Odd Number 5
Odd Number 7
Odd Number 9
Even Number 0
Even Number 2
Even Number 4
Even Number 6
Even Number 8
Even Number 10
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$
```

5. Write a program to demonstrate wait and notify methods.

Answer: Filename- **WaitNotify.java**

```
Terminal: Local × Local (2) × +
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ javac WaitNotify.java
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ java WaitNotify

Waiting for the thread to complete
100
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$
```

6. Write a program to demonstrate sleep and join methods.

Answer: Filename- **SleepJoin.java**

```
Terminal: Local × Local (2) × +
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ javac SleepJoin.java
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ java SleepJoin
1
2
3
4
1
2
3
4
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$
```

7. Run a task with the help of callable and store it's result in the Future.

Answer: Filename- **CallableFuture.java**

```
Terminal: Local x Local (2) x +
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ javac CallableFuture.java
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ java CallableFuture
java.io.IOException:
Sleeping for too long.
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$
```

8. Write a program to demonstrate the use of semaphore.

Answer: Filename- **SemaphoreUse.java**

```
Terminal: Local x Local (2) x +
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ javac SemaphoreUse.java
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$ java SemaphoreUse
Starting B
Starting A
B is waiting for a permit.
A is waiting for a permit.
B gets a permit.
B: -1
B: -2
B: -3
B: -4
B: -5
B releases the permit.
A gets a permit.
A: -4
A: -3
A: -2
A releases the permit.
count: -2
vishal@Vishal-Sharma:~/Desktop/Multithreading in JAVA$
```

9. Write a program to demonstrate the use of CountDownLatch

Answer: Filename- **CountDownLatchUse.java**

