

Basics Of Multi-threading in Java

Name - Prashant Brahmabhatt

ID - 4092

Email - prashant.brahmbhatt@tothenew.com

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

Volatile.java



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 5: Java
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ /usr/lib/jvm/java-11-openjdk-amd64/bin/
UTF-8 -cp /home/prashant/.config/Code/User/workspaceStorage/6c095099dc18527dc2b0248982590336/redhat
TN_2020_5cd602dd/bin exercises.Volatile
Thread 12: Old Value = 0
Thread 13: Old Value = 1
Thread 12: New Value = 3
Thread 13: New Value = 5
Thread 12: Old Value = 6
Thread 13: Old Value = 7
Thread 12: New Value = 9
Thread 13: New Value = 11
Thread 12: Old Value = 12
Thread 13: Old Value = 13
Thread 12: New Value = 15
Thread 13: New Value = 17
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$
```

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

RunnableUse.java

```
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ cd /home/prashant/BootcampAss/Bootcamp_TTN_2020 ; /usr/lib/j
vm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home/prashant/.config/Code/User/workspaceStorage/6c095099dc18
527dc2b0248982590336/redhat.java/jdt_ws/Bootcamp_TTN_2020_5cd602dd/bin exercises.RunnableUse
Inside thread class in anonymous 0
Inside thread class in Runnable 0
Inside thread class in anonymous 1
Inside thread class in Runnable 1
Inside thread class in anonymous 2
Inside thread class in Runnable 2
Inside thread class in anonymous 3
Inside thread class in Runnable 3
Inside thread class in anonymous 4
Inside thread class in Runnable 4
Inside thread class in anonymous 5
Inside thread class in Runnable 5
Inside thread class in anonymous 6
Inside thread class in Runnable 6
Inside thread class in anonymous 7
Inside thread class in Runnable 7
Inside thread class in anonymous 8
Inside thread class in Runnable 8
Inside thread class in anonymous 9
Inside thread class in Runnable 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
```

3. Write a program using synchronization block and synchronization method

Sync.java

```
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home/prashant/.config/Code/User/workspaceStorage/6c095099dc18527dc2b0248982590336/redhat.java/jdt_ws/Bootcamp_TTN_2020_5cd602dd/bin exercises.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 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
prashant@prashant:Bootcamp_TTN_2020 (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.

OddEven.java

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Java Proc
encoding=UTF-8 -cp /home/prashant/.config/Code/User/workspaceStorage/6c095099dc18527dc2b0248
/jdt_ws/Bootcamp_TTN_2020_5cd602dd/bin exercises.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
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$
```

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

WaitNotify.java

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ cd /home/prashant/Bootca
bin/java -Dfile.encoding=UTF-8 -cp /home/prashant/.config/Code/User/workspaceStorage/6
TN_2020_5cd602dd/bin exercises.WaitNotify

Waiting for the thread to complete
100
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$
```

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

SleepJoin.java

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ cd /home/prashant/Bootcamp
bin/java -Dfile.encoding=UTF-8 -cp /home/prashant/.config/Code/User/workspaceStorage/6c
TN_2020_5cd602dd/bin exercises.SleepJoin
1
2
3
4
1
2
3
4
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ █
```

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

CallableFuture.java

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Java Process Console v

prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ cd /home/prashant/BootcampAss/Bootcamp_TTN_2020 ; /u
vm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home/prashant/.config/Code/User/workspaceStorage/6c05
527dc2b0248982590336/redhat.java/jdt_ws/Bootcamp_TTN_2020_5cd602dd/bin exercises.CallableFuture

Starting ...
Finished.
Result is: 1309

prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ cd /home/prashant/BootcampAss/Bootcamp_TTN_2020 ; /u
vm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home/prashant/.config/Code/User/workspaceStorage/6c05
527dc2b0248982590336/redhat.java/jdt_ws/Bootcamp_TTN_2020_5cd602dd/bin exercises.CallableFuture
java.io.IOException:
Sleeping for too long.
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ █
```

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

SemaphoreUse.java

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Java Process Console v + []

prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ cd /home/prashant/BootcampAss/Bootcamp_TTN_2020 ; /usr/lib/j
vm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home/prashant/.config/Code/User/workspaceStorage/6c095099dc18
527dc2b0248982590336/redhat.java/jdt_ws/Bootcamp_TTN_2020_5cd602dd/bin exercises.SemaphoreUse
Starting A
A is waiting for a permit.
Starting B
B is waiting for a permit.
A gets a permit.
A: 1
A: 2
A: 3
A releases the permit.
B gets a permit.
B: 2
B: 1
B: 0
B: -1
B: -2
B releases the permit.
count: -2
prashant@prashant:Bootcamp_TTN_2020 (Multithreading_in_Java)$ █
```

