### **GAM/IT/2022/F/0084 - R.D.S. NISANSALA**

#### ASSIGNMENT – JAVA THREAD

## 1. Create a Simple Thread Class

```
public class SimpleThread extends Thread {
         @Override
         public void run() {
         System.out.println(Thread.currentThread().getId() + " is executing the thread.");
         public static void main(String[] args) {
                                        SimpleThread thread1 = new SimpleThread();
                                        SimpleThread thread2 = new SimpleThread();
                                        thread1.start(); // Starts thread1
                                        thread2.start(); // Starts thread2
         File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
         🚰 🚰 🤚 : 🍏 🧖 : <default config>
                                                                                                    V → T 👸 → . 65 🚯 - 🕞 - 12999/5580MBI 🗘 🗘
                                                  rojects × Files
          InventorySystem

JDBCExample
                                                                                             Source History | [6] | [6] - | [8] - | [7] | [7] - | [7] | [8] | [8] | [8] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | [9] | 
                                                                                          JBBCExample

a Source Packages

i jobcexample

b DatabaseConnection.java

Employee.java

p JBBCExample.java
             Test Packages
          Test Libraries

MultiThreadApp

Source Packages

multithreadapp
                         😷 🚰 🚰 🤚 🤚 (46fault config> 🗸 📦 - 🔐 🎉 - 🍇 🚯 - 🔞 - 🔞 - 4562/5580MB 🕻 🗘
                                                __ Start Page × 🚳 MultiThreadApp,java × 💆 SimpleThread.java ×
         oiects × Files
          InventorySystem
JDBCExample
                                                                                               JDBCExample

DatabaseConnection.java
Employee.java
EmployeeDAO.java
JDBCExample.java
                                                                                                           package multithreadapp;
                                                                                                          public class MultiThreadApp (
                                                                                                                public static void main(String[] args) (
    SimpleThread thread! = new SimpleThread();
    SimpleThread thread2 = new SimpleThread();
    thread1.start(); // Starts thread2
    thread2.start(); // Starts thread2
           Test Packages
                Libraries
                Test Libraries
         ■ Test Libraries

MultThreadApp

■ Source Packages

■ multihreadapp

■ Counter_java

⑤ MultThreadApp_java

② RunnableTask,java

② SimpleThread.java
                                                                                                                        /* EunnableTask task1 = new EunnableTask();

RunnableTask task2 = new EunnableTask();

Thread thread1 = new Thread(task2);

Thread thread2 = new Thread(task2);

thread3.start(); // Starts thread1

thread2.start(); // Starts thread2*/

Simple Inread Java

Task java

Task java

Thread Lifecycle Example java

Test Packages

Libraries
                  15 is executingthe thread.
                  16 is executing the thread.
BUILD SUCCESSFUL (total time: 0 seconds)
```

## 2. Using Runnable Interface

```
public class Runnable Task implements Runnable {
@Override
public void run() {
       System.out.println(Thread.currentThread().getId() + " is executing the runnable task.");
public static void main(String[] args) {
      RunnableTask task1 = new RunnableTask();
      RunnableTask task2 = new RunnableTask();
      Thread thread 1 = \text{new Thread}(\text{task 1});
      Thread thread2 = \text{new Thread}(\text{task2});
      thread1.start(); // Starts thread1
      thread2.start(); // Starts thread2
 🕾 🚰 🛂 🤚 : 🤚 (# <default config> 🗸 📦 - 🚏 🎇 👂 - 🍇 🚮 - 🕦 - : [_340.6/558.0M8] 📞 🗘
                        ___ Start Page × 🍇 MultiThreadApp,java × 🧧 SimpleThread.java × 🧧 RunnableTask,java ×
Projects × Files
 InventorySystem

JDBCExample

Source Packages

Jdbcexample
                                                Jibbesample

BatabaseConnection,java

Employee,java

Employee)AO,java

DBCEkample,java

Jiberses

Jiberses

MuttThreadapp

Source Packages
                                                       public class RunnableTask implements Runnable (
                                                        GOverride
                                                                  public void run() (
                                                                  System.out.println(Thread.currentThread().getId() + " is executingthe runnable task.");
      multithreadapp
multithreadapp

Gounter, Java
Multi ThreadApp, Java
Runnable Task, Java
Synchronized Example, Java
ThreadUlecycleExample, Java
ThreadUlecycleExample, Java
ThreadDeolExample, Java
  > Test Packages
Q File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help MultiThreadApp - Apache NetBeans IDE 17
 __ Start Page × 🚳 MultiThreadApp.java × 🚳 SimpleThread.java × 🚳 RunnableTask.java ×
Projects × Files
  InventorySystem
JDBCExample
                                                 Source Packages

ightharpoonup jdbcexample
                                                       package multithreadapp;
 Jidboexample

DatabaseConnection.java

EmployeeJava

EmployeeJAO.java

Jidboexample.java

Jidboexample.java

Jidboexample.java

Jidboexample.java

Jidboexample.java

Jidboexample.java

Jidboexample.java

Jidboexample.java
                                                       public class MultiThreadApp {
                                                           public static void main(String[] args) {
    /*SimpleThread thread1 = new SimpleThread();
    SimpleThread thread2 = new SimpleThread();
    thread1.stat(); // Starts thread1
    thread2.stat(); // Starts thread2*/
     Source Packages

multithreadapp
                                                       RunnableTask task1 = new RunnableTask();
RunnableTask task2 = new RunnableTask();
Thread thread1 = new Thread(tasps:task1);
Thread thread2 = new Thread(tasps:task2);
thread1.start(); // Starts thread1
thread2.start(); // Starts thread3
multithreadapp

© Counter java

Multi ThreadApp java

Runnable Taskjava

Simple ThreadJava

SynchronizedExample java

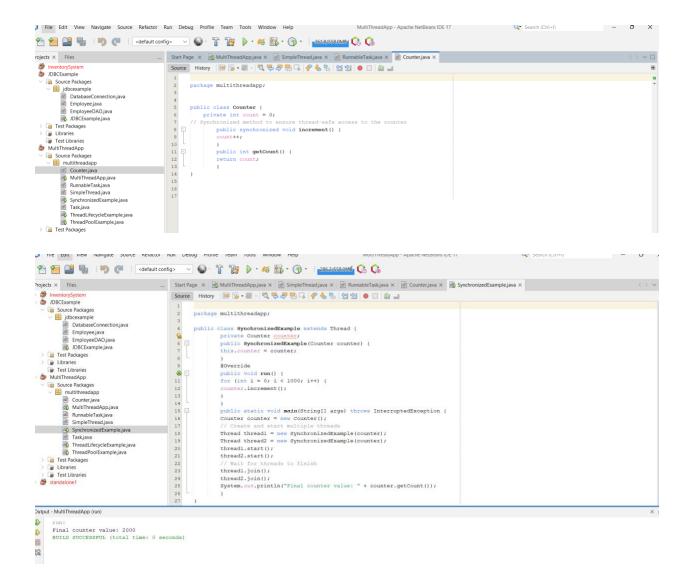
Taskjava

ThreadUcoExample java

ThreadDooExample java
   > iii Test Packages
     Libraries
Output - MultiThreadApp (run)
```

# 3. Synchronizing Threads

```
class Counter {
private int count = 0;
// Synchronized method to ensure thread-safe access to the counter
public synchronized void increment() {
count++;
public int getCount() {
return count;
public class SynchronizedExample extends Thread {
private Counter counter;
public SynchronizedExample(Counter counter) {
this.counter = counter;
@Override
public void run() {
for (int i = 0; i < 1000; i++) {
counter.increment();
public static void main(String[] args) throws InterruptedException {
Counter counter = new Counter();
// Create and start multiple threads
Thread thread1 = new SynchronizedExample(counter);
Thread thread2 = new SynchronizedExample(counter);
thread1.start();
thread2.start();
// Wait for threads to finish
thread1.join();
thread2.join();
System.out.println("Final counter value: " + counter.getCount());
```



## 4. Using ExecutorService for Thread Pooling

```
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
class Task implements Runnable {
private int taskId;
public Task(int taskId) {
this.taskId = taskId;
@Override
public void run() {
System.out.println("Task " + taskId + " is being processed by " +
Thread.currentThread().getName());
public class ThreadPoolExample {
public static void main(String[] args) {
// Create a thread pool with 3 threads
ExecutorService executorService = Executors.newFixedThreadPool(3);
// Submit tasks to the pool
for (int i = 1; i \le 5; i++) {
executorService.submit(new Task(i));
// Shutdown the thread pool
executorService.shutdown();
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help MultiThreadApp - Apache NetBeans IDE 17
 _ Start Page X 🚳 MultiThreadApp.java X 🚳 SimpleThread.java X 🚳 RunnableTask.java X 🚳 Counter.java X 🚳 SynchronizedExample.java X
 Projects X files

InventorySystem

IDBCExample

DatabaseConnection.java

Employee:Java

Employee:Java

IDBCExample java
                                package multithreadapp;

import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
                                 Source History 🔑 🖫 - 🔍 🕏 🗗 🖫 🔗 🦠 🧐 😅 😅 📵 🗆 🕮 🚅
                                 public class ThreadPoolExample (
public static void main(Strir
// Create a thread pool with
the ExecutorService executorServi
// Submit tasks to the pool
for (int i = 1; i <= 5; i++)
executorService.submit(new Ta
)
// Shuddown the thread pool
executorService.shutdown();
)

22
                                        public static void main(String[] args) {
// Create a thread pool with 3 threads
       SynchronizedExample.java
Task.java
```

## 5. Thread Lifecycle

```
public class ThreadLifecycleExample extends Thread {
@Override
public void run() {
    System.out.println(Thread.currentThread().getName() + " - State: " +
    Thread.currentThread().getState());
try {
    Thread.sleep(2000); // Simulate waiting state
    } catch (InterruptedException e) {
    e.printStackTrace();
    System.out.println(Thread.currentThread().getName() + " - State after
    sleep: " + Thread.currentThread().getState());
public static void main(String[] args) {
    ThreadLifecycleExample thread = new ThreadLifecycleExample();
    System.out.println(thread.getName() + " - State before start: " +
    thread.getState());
    thread.start(); // Start the thread
    System.out.println(thread.getName() + " - State after start: " +
    thread.getState());
      FILE EDIT VIEW NAVIGATE SOURCE RETACTOR KUN LJEDUG PROTILE LEARN LOOIS WINDOW HELD
    _ __va 

SimpleThreadJava × 

RunnableTaskjava × 

Counter_java × 

SynchronizedExamplejava × 

ThreadPoolExamplejava × 

ThreadDifecycleExamplejava ×
                                      Source History 🐼 🖟 🚚 - 🔼 👺 🕾 👺 🥞 😂 💇 👲 📵 🔲 🕮 🚅
    package multithreadapp;
                                            public class ThreadLifecycleExample extends Thread (
                                                  ge
public void run() {
   System.out.println(Thread.currentThread().getName() + " - State; " +
   Thread.currentThread().getState());
                                                 try (
Thread.sleep(millis:2000); // Simulate waiting state
     MultiThreadApp
       Source Packages

Source Packages

multithreadapp

Counterjava

MultiThreadAppjava

RunnableTaskjava

SimpleThreadjava

SynchronizeExample.java

TreaddfecycleExample.java
                                                 ) catch (InterruptedException e) {
e.printStackTrace();
                                                   public static void main(String[] args) {
                                                  ThreadLifecycleExample thread = new ThreadLifecycleExample();
System.out.println(thread.getName() + " - State before start: "
                                                   thread.getstate(); // Start the thread
System.out.println(thread.getName() + " - State after start: " +
thread.getState());
      Test Packages
Libraries
Test Libraries
    output - MultiThreadApp (run)
        Thread-0 - State before start: NEW
Thread-0 - State after start: RUNNABLI
Thread-0 - State: RUNNABLE
Thread-0 - State aftersleep: RUNNABLE
        BUILD SUCCESSFUL (total time: 2 seconds)
```