**Operating Systems and Concurrency**

**Student Name:** Raphael Salaja

**AIT Student number:** A00269349

**Assignment:** Assignment 2

**Date of Submission:** 13/10/2021

**FORM A1**

# STUDENT PLAGIARISM DISCLAIMER FORM

****

## PLAGIARISM DISCLAIMER

STUDENT NAME: Raphael Salaja

STUDENT NUMBER: A00269349

PROGRAMME: BSc (Hons) in Software Design with Virtual Reality and Gaming

YEAR: 3

MODULE: Operating Systems and Concurrency

LECTURER: Thiago Braga Rodrigues

ASSIGNMENT TITLE: Lab 2 Assignment

DUE DATE: 20 October 2021

DATE SUBMITTED: 13 October 2021

ADDITIONAL INFORMATION:

I understand that plagiarism is a serious academic offence, and that AIT deals with it according to the AIT Policy on Plagiarism.

I have read and understand the AIT Policy on Plagiarism and I agree to the requirements set out therein in relation to plagiarism and referencing. I confirm that I have referenced and acknowledged properly all sources used in preparation of this assignment. I understand that if I plagiarise, or if I assist others in doing so, that I will be subject to investigation as outlined in the AIT Policy on Plagiarism.

I understand and agree that plagiarism detection software may be used on my assignment. I declare that, except where appropriately referenced, this assignment is entirely my own work based on my personal study/or research. I further declare that I have not engaged the services of another to either assist in, or complete this assignment.

## Signed:Raphael Salaja

**Dated: 13 October 2021**

# Contents

[Classes 4](#_Toc85026586)

[1. ATM Simulator Class 4](#_Toc85026587)

[2. Account Class 5](#_Toc85026588)

[Before Synchronization 5](#_Toc85026589)

[After Synchronization 6](#_Toc85026590)

[3. Bank Class 7](#_Toc85026591)

[4. Company Class 8](#_Toc85026592)

# Classes

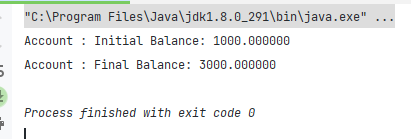
## ATM Simulator Class

*/\*\*  
 \** ***@author*** *Raphael Salaja  
 \** ***@version*** *13/10/21  
 \*/*public class ATM\_Simulator {  
 public static void main(String[] args) throws InterruptedException {  
 Account account = new Account();  
 account.setBalance(1000);  
  
 Company company = new Company(account);  
 *Runnable* runnable\_company = company;  
 Thread thread\_company = new Thread(runnable\_company);  
  
 Bank bank = new Bank(account);  
 *Runnable* runnable\_bank = bank;  
 Thread thread\_bank = new Thread(runnable\_bank);  
  
 System.*out*.printf("Account : Initial Balance: %f\n", account.getBalance());  
  
 thread\_company.start();  
 thread\_bank.start();  
  
 thread\_company.join();  
 thread\_bank.join();  
  
 System.*out*.printf("Account : Final Balance: %f\n", account.getBalance());  
  
 }  
}

## Account Class

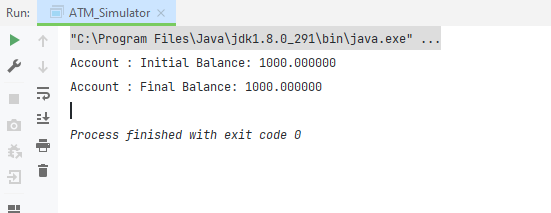
### Before Synchronization

*/\*\*  
 \** ***@author*** *Raphael Salaja  
 \** ***@version*** *13/10/21  
 \*/*public class Account {  
 private double balance;  
  
 public Account() {  
 }  
  
 public double getBalance() {  
 return balance;  
 }  
  
 public void setBalance(double balance) {  
 this.balance = balance;  
 }  
  
 public void addAmount(double amount) {  
 double tmp = balance;  
 setBalance(tmp + amount);  
 try {  
 Thread.*sleep*(10);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void subtractAmount(double amount) {  
 double tmp = balance;  
 setBalance(tmp - amount);  
 try {  
 Thread.*sleep*(10);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }



### After Synchronization

*/\*\*  
 \** ***@author*** *Raphael Salaja  
 \** ***@version*** *13/10/21  
 \*/*public class Account {  
 private double balance;  
  
 public Account() {  
 }  
  
 public synchronized double getBalance() {  
 return balance;  
 }  
  
 public synchronized void setBalance(double balance) {  
 this.balance = balance;  
 }  
  
 public synchronized void addAmount(double amount) {  
 double tmp = balance;  
 setBalance(tmp + amount);  
 try {  
 Thread.*sleep*(10);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public synchronized void subtractAmount(double amount) {  
 double tmp = balance;  
 setBalance(tmp - amount);  
 try {  
 Thread.*sleep*(10);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
}



## **Bank Class**

*/\*\*  
 \** ***@author*** *Raphael Salaja  
 \** ***@version*** *13/10/21  
 \*/*public class Bank implements *Runnable* {  
 private final Account account;  
 public Bank(Account account) {  
 this.account = account;  
 }  
 @Override  
 public void run() {  
 for (int i = 0; i < 100; i++) {  
 account.subtractAmount(1000);  
 }  
 }  
}

## Company Class

*/\*\*  
 \** ***@author*** *Raphael Salaja  
 \** ***@version*** *13/10/21  
 \*/*public class Company implements *Runnable* {  
 private final Account account;  
 public Company(Account account) {  
 this.account = account;  
 }  
  
 @Override  
 public void run() {  
 for (int i = 0; i < 100; i++) {  
 account.addAmount(1000);  
 }  
 }  
}