CS251: Intro to Software Engineering Homework 2 (4 marks)

Due Date 26 April 2021 @ 11:00 pm (submit on Blackboard)

TA Name: …… Mark: ……..

CS251 –Introduction to Software Engineering, 2021 Each student fills this form for his program and gives it to TA

The questions to answer about each program are included in the following form. **Print and fill this form and bring with you to the discussion.**

Student name: Peter Atef Nagiub **ID:** 20190140 **Group:**2

1. **Which Program (Idea for task 2) did you choose?**

We made a client handler class to handle operations on a specific client, we did that so that client handling is separated from the main method and to make the code cleaner and more readable

Which of the following Java / OOP features did you use in your program?

2. **How many classes did you create and their names?**

   I made Account, Special Account and an AccountMain class to test all the functionalities

3. **How many different access specifiers did you use and their names?**

   Public and protected and private

4. **How many Java coding style rules did you use and which ones?**

   I used camelCase to write clear and meaningful variable and method names

5. **How many Javadoc tags did you use and which ones?**

   I used 5 Javadoc tags which are @author, @param, @return, @throws and @since

6. **Did you use inheritance? When and why?**

   Yes, I used inheritance to extend the Account class to a Special Account class to enable overdrafting by -1000

7. **Did you use method overriding? When and why?**

   Yes, I overrode the toString method to return a meaningful string for the current Account instance and I also overrode the setBalance and withdraw methods in the Special Account Class to enable the balance to go below zero and until -1000

8. **Did you use method composition? When and why?**

   No

9. **Did you use method polymorphism? When and why?**

   Yes, I used polymorphism in AccountMain class because I wanted to create an instance of Account class as an instance of Special Account class

10. **Draw in the space below a simple UML class diagram that shows your main classes, their attributes and operations and their interactions with each other.**