**OOD Lab 4 – Abstract Classes/Static Members**

* Put your name and student number in a comment at the top of each file.
* Ensure that your code is well formatted e.g. lines after an opening brace should be indented. Each closing brace should be at the same indentation as the statement which opened it. (Use Ctrl + Shift + F if using Eclipse)
* It is compulsory to comment your code appropriately. For each of the exercises ensure that your code is sufficiently commented to demonstrate your understanding of the software you are writing. You do not have to comment every line however.
* Marks will be lost if the above instructions are not followed.
* You may discuss assignments with you classmates but you must write each assignment yourself. Copying another student’s code will be considered plagiarism. Outcomes vary from receiving zero in assessments to failing the course.

1. From your previous lab work, update the **Account**, **SavingsAccount**, and **CurrentAccount** classes as follows:
   * 1. Make the **Account** class an abstract class. Define an abstract method **printAccountInfo**() which should be implemented by the sub classes. This method should display the type of account along with the account details.
     2. Define a class variable in each sub class (**SavingsAccount** and **CurrentAccount**), which will keep a track of how many objects have been created from that class.
     3. Create a static method in the **SavingsAccount** and **CurrentAccount** classes called **displayNumberOfAccounts**(). This method should show the number of accounts created of that particular sub class.
2. Update the **Bank** class in order to test the updates that you have made. Create 2 objects for each subclass. Using the objects call the above methods to illustrate that they are functioning correctly.

**What do I submit?**

Submit the Account.java, SavingsAccount.java, CurrentAccount.java, and Bank.java files on Moodle.