Sheet 3: Static, Overloading, Constructor, Composition

1. **From sheet2 Q1: update the bank account class with the following:**

   a) **Overload the constructor by the following:**

   - **Add new constructor with three parameters: account number, account name, balance**

   b) **Add a "Transfer" method which will get money from one account and put it into another account, the method will take parameters :**

   - **From account**
   - **To account**
   - **Amount**

   c) **"Transfer" method in point (b) should be instance or static?**

   1.2. **Write a Test Class which perform the following:**

   a) **create savingAccount bank accounts objects with 5000 as initial balance and your name**

   b) **create currentAccount bank accounts objects with 0 as initial balance**

   c) **update currentAccount with your brother name**

   d) **transfer 300 from savingAccount to currentAccount**

   e) **finally display updated info for each account as following:**

   - **account name/account number => balance**

2. **Write a class to handle a Facebook of AzharWall page which allow comments from students created from a class called Student:**

   2.1. **Use static variable wall in AzharWall class**

   2.2. **Every Student object should have name (instance variable)**

   2.3. **Make every Student append his message it to wall variable**

   2.4. **When student append message, it should be in the following format:**

   a) **name : message**

   b) **Example : "Hosam :Hi" where "Hosam" is the name & "Hi" is the message appended**

2.5. **Keep trace of No. of object created from student class**

2.6. **The student class should have a method to check No. of instance created**

2.7. **every student object should be able to see all messages written on AzharWall using viewWall method which will display chat static variable + the total number of created object**

2.8. **Write the main method that test that class with creating at least three student objects, and each object should write two messages and make them display wall after each object finish talk**

3. **Write a Complex class for complex number which contains the following:**

3.1. **Three constructors:**

   a) **With zero parameters: which initialize real=0 and imaginary=0**

   b) **With two floating number as parameters: aReal and aImaginary**

   c) **With Compelled object as parameters: c**

3.2. **Two overloaded adding methods**

   a) **First one to Add float to the complex object (to real part)**

   b) **Second one to Add complex to the complex object (real to real and imaginary to imaginary)**

3.3. **Two overloaded subtract methods same as add**

3.4. **Two overloaded multiply methods same as add (notes multiply in float will multiply in both real and imaginary)**

3.5. **Display method to display object real and imaginary as following:**

   a) **"real + j imaginary" such as : 5 + j3**

3.6. **Create TestComplex class which should test all above as following:**

   a) **Create three objects, each object with different constructor**

   b) **Add float number to first object**

   c) **Add first object on second object**

   d) **Subtract float number from second object**

e) **Subtract second object from third object**

f) **Multiply first object by float number**

g) **Multiply first object by third object**

h) **Display three objects**

4. **Write an email program which contains the following classes**

   4.1. **Message class which contain three objects from three different classes as following:**

   - **From object: from Person class**
   - **To object: from Person class**
   - **content: from MessageContent class**
   - **display() method: to display message content**

   4.2. **Person class should contains the following:**

   a) **Name: as string**

   b) **Email: as string**

   4.3. **MessageContent class should contains the following:**

   a) **subject: as string**

   b) **body: as string**

   4.4. **write a test program which compose a message contain "Next month there is surprise for student, there is OOP exam" where from is Hosam (hosam@azhar.edu.eg) & to is Rohim(rohim@azhar.edu.eg)**

   4.5. **then display the message object with the following format:**

   From: name (email)

   To: name (email)

   Body: message

5. **Homework (to be delivered as report)**

   5.1. **for the project you selected please start implement it as following:**

   a) **update your project classes with the following:**

   - **static variables or method as needed**
   - **at least two constructor for each class you have**
   - **add overloaded method as needed**
   - **build relation of "has-a" between classes**