

**NANYANG TECHNOLOGICAL UNIVERSITY**  
**SEMESTER 2 EXAMINATION 2016-2017**  
**CE2002/CZ2002 – OBJECT ORIENTED DESIGN & PROGRAMMING**

Apr/May 2017

Time Allowed: 2 hours

**INSTRUCTIONS**

1. This paper contains 4 questions and comprises 9 pages.
  2. Answer **ALL** questions.
  3. This is a closed-book examination.
  4. All questions carry equal marks.
  5. APPENDIX A shows a Class Hierarchy referenced by Question 1.
  6. APPENDIX B shows a Class Hierarchy referenced by Question 2.
  7. APPENDIX C shows the Class Diagram referenced by Question 3(a).
  8. APPENDIX D shows the Sequence Diagram referenced by Question 3(b).
- 

1. (a) Explain what is “Object”, as one of the four basic components of the Object-Oriented Model.  
(3 marks)
- (b) On top of “Object”, list the other THREE basic components of the Object-Oriented Model.  
(3 marks)
- (c) List and explain four cases of naming convention in the Java language.  
(4 marks)

Note: Question No. 1 continues on Page 2

- (d) Study the class hierarchy diagram in Appendix A (page 6). Identify the class whose moveIt (...) or feedIt (...) method will be called or the type of error they may encounter for each of the following SIX sets of statements :
- (i) InterfaceL z1 = new ClassBP();  
z1.moveIt(new ClassB());
- (ii) InterfaceL z1 = new ClassBP();  
ClassB z2 = (ClassB)z1;  
z2.moveIt(new ClassB());
- (iii) ClassBP z1 =new ClassBP();  
z1.feedIt(new ClassV(), new ClassM());
- (iv) ClassM z1 = new ClassB();  
z1.feedIt(new ClassV());
- (v) ClassM z1 = new ClassB();  
z1.feedIt(new ClassBP());
- (vi) InterfaceL z1 = new ClassBP();  
ClassP z2 = (ClassP)z1;  
z2.moveIt(new ClassV());

(15 marks)

2. The Alicaca e-commerce platform has an hourly sale function, which selects a product for discounted sale for a limited time of one hour. You will write the code of this hourly sale function in a Java class, called HourlySaleApp. In addition to this class, there are two other classes, which you will also write code for, i.e. Sale and DiscountSale. The class diagram for the classes of Sale and DiscountSale are shown in Appendix B (page 7).

Each Sale object holds the name (name) of the sale, and the price of the sale (price).

The Sale objects have the following behavior:

- Sale (name: String, price: double) is the constructor for the class which initializes the name and the price.
- getPrice() is a get method that returns the price.

Note: Question No. 2 continues on Page 3

- `lessThan(anotherSale : Sale)` is a method that compares this sale with another sale and returns true if the price of the sale is less than that of another and false if otherwise.
- `bill()` is a method that prints out the price.

`DiscountSale` object holds the attribute `discount` which is the percentage of the original price after discount. For example, if the price is 30% off, the value of `discount` attribute is 0.7.

The `DiscountSale` objects have the following behaviors in addition:

- `DiscountSale(n: String, p: double, d: double)` is the constructor for the class which initializes the name, the normal price and the discount of the sale respectively.
- `bill()` is a method that prints out the discounted price of the discount sale, which is the normal price multiplied by the discount percentage.

The `HourlySaleApp` object has a main method. In the main method, a `Sale` object reference (`normalSale`) and a `DiscountSale` object reference (`discountSale`) are declared. They are made pointing to newly created objects of `Sale` and `DiscountSale` for the same product of “Samsung Galaxy Note 7” with the original price of \$600.00 respectively. For the object of `DiscountSale`, the discount is 0.5. The price of the `normalSale` is checked if it is **less than** that of the `discountSale`. The result is then printed out.

(25 marks)

3. (a) The UML **Class Diagram** in Appendix C (page 8) shows the relationships of THREE classes: `Fraction`, `Vector2D` and `Vector3D`. Study the class diagram and the details depicted carefully.

Additional details relevant to this question are provided below.

The `Fraction` class has 2 integer attributes, `num` and `deno`, which store the numerator and denominator of a fraction respectively. The constructor initializes the `num` and `deno` attributes. The `print` function displays the value of `num` and `deno` in the format “`<num>/<deno>`”. The `operator+` function overloads the “`+`” operator to add 2 fractions together by adding the input fraction to itself and returns the resultant fraction.

Note: Question No. 3 continues on Page 4

The Vector2D class has 2 Fraction attributes, `x` and `y`, which store the x-axis and y-axis values of a two-dimensional vector respectively. There are 2 constructors; one can be used to initialize the `x` and `y` fractions directly and the other can be used to initialize `x` and `y` fractions by using the Fraction class constructor. The print function displays the value of `x` and `y` in the format “`<num>/<deno>i <num>/<deno>j`”. The `operator+` function overloads the “`+`” operator to add 2 two-dimensional vector together by adding the input vector to itself and returns the resultant vector.

The Vector3D class inherits from the Vector2D class and has a Fraction attribute, `z`, which stores the z-axis value of a three-dimensional vector. Each of the 2 constructors uses its base class corresponding constructor to initialize the `x` and `y` fractions and also initialize the `z` fraction directly or uses the Fraction class constructor. The print function displays the value of `x` and `y` in the format “`<num>/<deno>i <num>/<deno>j <num>/<deno>k`”. The `operator+` function overloads the “`+`” operator to add 2 three-dimensional vector together. **The `operator+` function checks the validity of the casting from Vector2D to Vector3D before adding the input vector to itself and returns the resultant vector. If casting failed, it returns itself.**

- (i) Write the C++ code for the Vector3D class in a header file, `vector3d.h`, as depicted in the class diagram and implement the 2 constructors. You can assume the header files for the Vector2D and Fraction class are `vector2d.h` and `fraction.h` respectively and you need not implement them.

(6 marks)

- (ii) Write the C++ code for the Vector3D class in a corresponding implementation file, `vector3d.cpp`, for the other functions. You should use the appropriate C++ keyword/s to ensure the code will run as expected.

(7 marks)

- (b) The UML **Sequence Diagram** in Appendix D (page 9) shows the objects' interaction of a scenario flow in a particular application. Using the details depicted in the diagram, write the preliminary JAVA code for the class Local and its methods. You may make appropriate assumptions on the method parameters, return types and return value(s) if they are not stated in the diagram.

(12 marks)

4. Consider a microblogging application, *Gossiper*, with the following requirements:

*Gossiper* has a number of users who each has a unique ID, joined date and a screen name. The screen name of a user can be changed and the application records the list of screen names that the user has used, as well as the starting and ending dates of usage of the name.

A user on *Gossiper* can follow any other user, and can post microblogs (referred to as “*sipes*”) onto *Gossiper*. A *sipe* posted by a user can be seen by all other users who follow him/her. The following relationships among users may change over time. *Gossiper* keeps track of the date when a user starts or stops following another user.

A *sipe* has a unique ID, a text that is *constraint* to at most 140 characters, number of likes and dislikes, and a post date/time. A *sipe* may also contain a number of photos as attachments and each photo has an ID, its size and storage location.

*Gossiper* allows a user to ‘like’ or ‘dislike’ a *sipe* and counts both accordingly. A user can file a complaint to *Gossiper* if he/she feels that a certain *sipe*’s content is inappropriate. Each complaint has a unique ID, a textual description, a filing date/time and a status. The status can be NEW, IN-PROGRESS or CLOSED. The complaint is handled by a *Gossiper* employee who will investigate and record the textual outcome when the complaint is settled with a settled date/time. An employee has an employee ID, rank and salary.

- (a) You are tasked to identify the entity classes needed to build the microblogging application based on the description above.

Show your design in a Class Diagram. Your Class Diagram should show clearly the relationship between classes, relevant attributes (at least TWO), logical multiplicities, meaningful role names, association names and constraint, if any. **You need not show the class methods.**

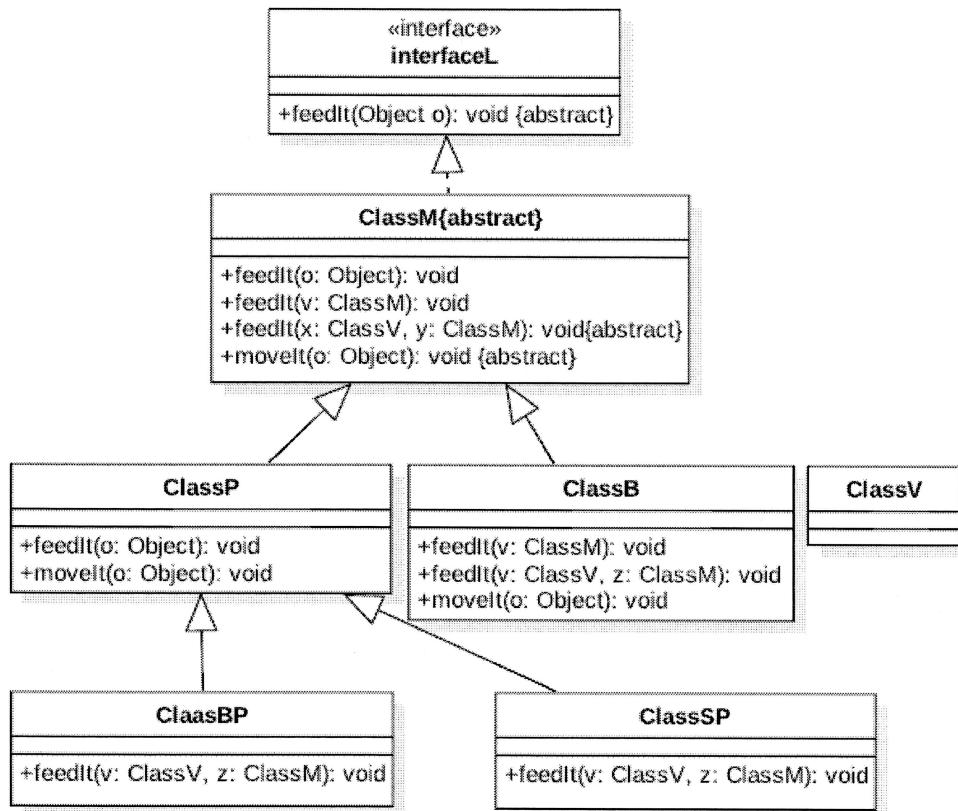
(15 marks)

- (b) Further to the above description, a user can also forward an existing *sipe* to his/her followers who are *Gossiper* users or non-*Gossiper* users via FaceBook, Twitter, WhatsApp, etc.

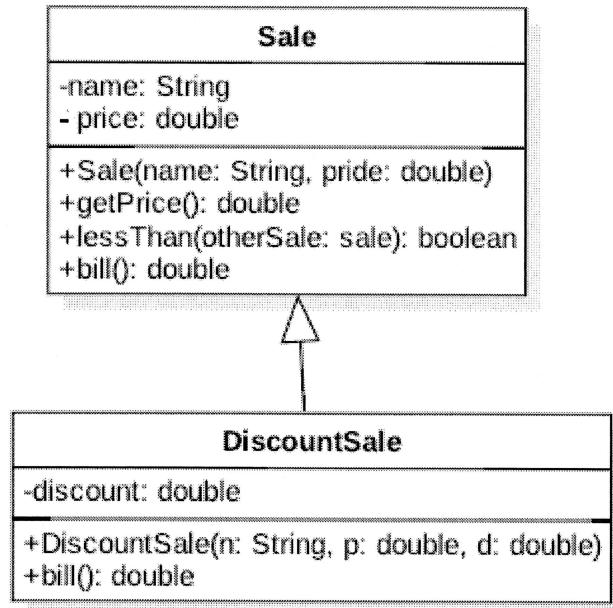
Suggest and explain, with a Class Diagram, how you can enable this requirement in your design with reusability, extensibility and maintainability in mind. State the design principle(s) you have applied. **You should show the class/interface method(s) to illustrate your idea.**

(10 marks)

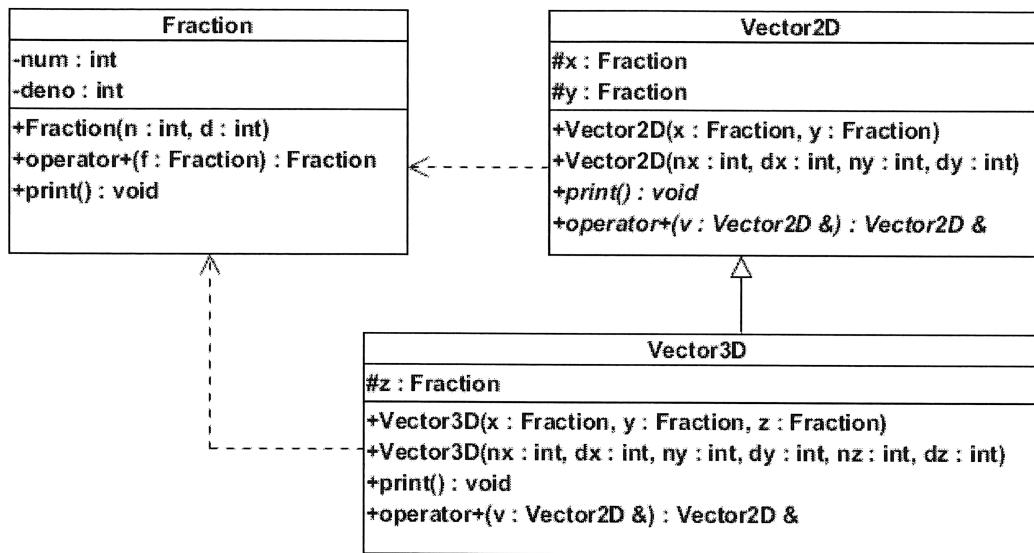
## APPENDIX A : Class Hierarchy



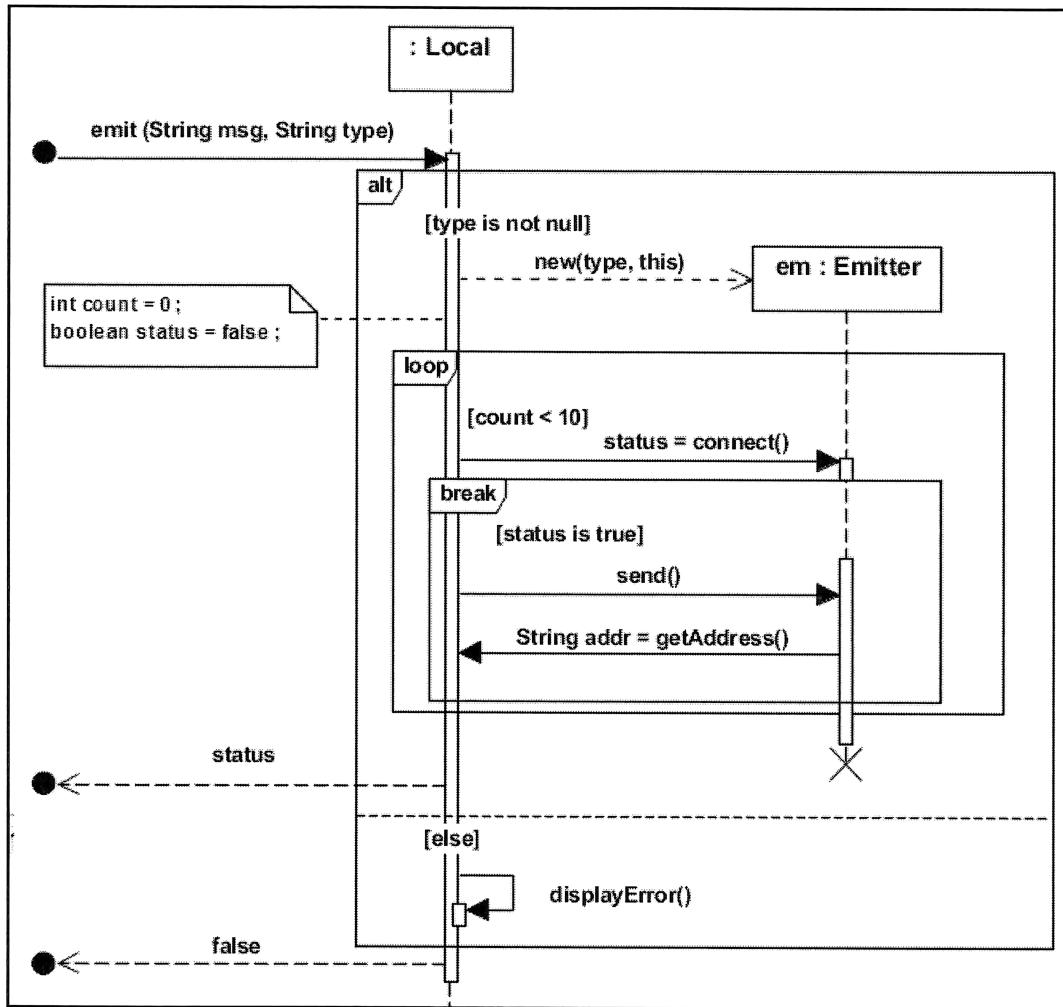
## APPENDIX B : Class Diagram



## APPENDIX C : Class Diagram



## APPENDIX D : Sequence Diagram



END OF PAPER

*ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library*

*ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library*

**CE2002 OBJECT ORIENTED DESIGN AND PROGRAMMING  
CZ2002 OBJECT ORIENTED DESIGN & PROGRAMMING**

Please read the following instructions carefully:

- 1. Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**
2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.
3. Please write your Matriculation Number on the front of the answer book.
4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.