

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2017-2018

CE2002/CZ2002 – OBJECT-ORIENTED DESIGN & PROGRAMMING

Nov/Dec 2017

Time Allowed: 2 hours

INSTRUCTIONS

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

-
1. (a) Explain what *Polymorphism* is in object oriented programming. What are the TWO benefits of polymorphism? (6 marks)
 - (b) Explain the concepts of *method overloading* and *method overriding*. Give THREE differences between the two concepts. (6 marks)
 - (c) Write Java code to demonstrate how to define a *static* method. And, list THREE differences between a *static* method and an *instance* method. (5 marks)

Note: Question No. 1 continues on Page 2

- (d) *Downcasting* will cause compilation error. How to avoid compilation error? Write Java code to demonstrate it. However, downcasting may still produce runtime error. How to avoid runtime error? Write Java code to demonstrate it. (8 marks)
2. Study the UML diagrams for class PhoneA and two interfaces IntPhoneA and IntPhoneB as shown in Figure Q2. Their methods function as follows:
- The method send (m: String) simply sends out the message m.
 - The method send(m: String, x: int) sends out the message m for x number of times.
 - The method send(m: String, n: String, x: int) sends out m and n as a single message for x number of times.
-
- ```
classDiagram
 class PhoneA {
 +send(m : String) : void
 }
 interface IntPhoneA {
 <<Interface>>
 +send(m : String) : void
 +send(m : String, x : int) : void
 }
 interface IntPhoneB {
 <<Interface>>
 +send(m : String, n : String, x : int) : void
 }
```
- Figure Q2**
- (a) Write Java code for the two interfaces IntPhoneA and IntPhoneB. (6 marks)
- (b) Assume that the PhoneA class has already been fully implemented. A class PhoneB inherits from the PhoneA class and implements both the interfaces IntPhoneA and IntPhoneB. The PhoneB class provides implementation for the send(m: String, x: int) method. Write Java code for the PhoneB class. (7 marks)
- (c) A class PhoneC inherits from the PhoneB class. Write Java code for the PhoneC class. (6 marks)
- (d) Draw the COMPLETE class diagram for all the classes and interfaces mentioned above, including all their methods. (6 marks)

3. (a) The UML **Class Diagram** in Appendix A (page 6) shows the relationships of TWO classes and TWO interfaces : Statement class, ArrayList class, Collection interface and List interface. ArrayList class, Collection interface and List interface are defined in a package called utility. Study carefully the class diagram and the details depicted.

Additional details relevant to this question are provided below.

The Statement class constructor initializes the words attribute by using the length parameter to create the ArrayList object using its constructor. The put function adds the word parameter to words list using the add function in the ArrayList class.

- (i) Write the C++ code for the Collection interface and List interface in a header file, **collection.h**, as depicted in the class diagram. The interfaces are to be defined in a C++ equivalent package.

(7 marks)

- (ii) Write the C++ code for the Statement class completely in an implementation file, **statement.cpp**. You can assume that the ArrayList class is also implemented in the **collection.h** header file.

(6 marks)

[You should use the appropriate C++ keyword/s to ensure the code will run as expected.]

- (b) The UML **Sequence Diagram** in Appendix B (page 7) 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 Consensus 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. (a) Consider a car racing event application, CREA, with the following requirements:

The car racing event is held every year. Each year's event consists of one

Note: Question No. 4 continues on Page 4

or more races. A race has the information of the location, the date, the road length and also the weather on the race date. The location will record the country, city and venue where the race is held.

One or more teams participate in each year's event. Each team has a team ID, the team name and the founding date of the team. Each team has one or more drivers participating in the each year's event. Each team has one or more members. The members include one manager, one or more drivers and one or more supporting staff. The member ID, the name, and the date of birth of member are recorded. Additionally, the driver's driving experience and past achievements are also recorded.

For each race, the time taken by each driver to finish the race is recorded. If the driver fails to finish a race, the cause of the unfinished race is recorded. The cause can be due to accidents or technical problems of his/her car. Every driver who finishes a race will be given a certain number of points based on his/her ranking (in terms of finish time) in the race.

After a year's races are finished, the team with the highest total points earned by its drivers during the year would be given the team championship of the year. The driver who earns the highest total points will be given an individual championship.

You are tasked to identify the entity classes needed to build the 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. For methods, you only need to show the method(s) which can be used to derive the individual and team championships.

(15 marks)

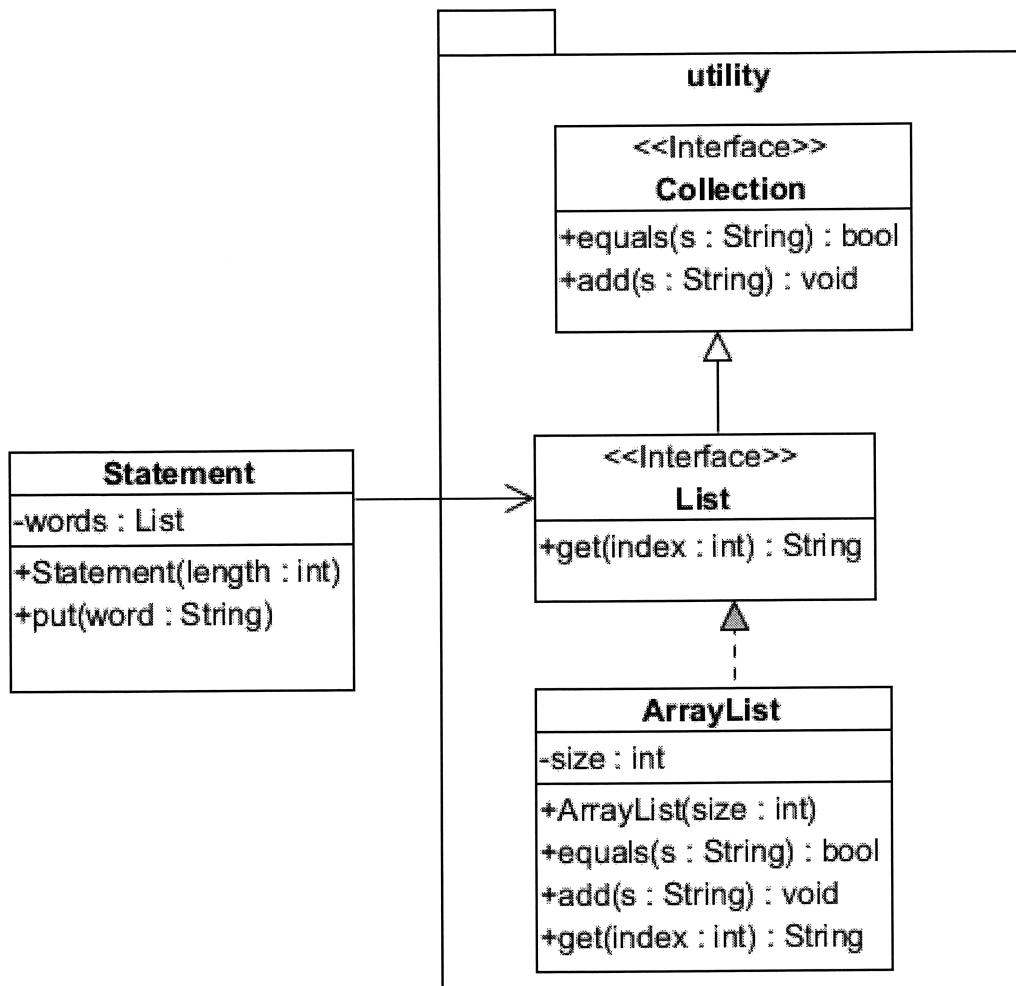
- (b) The UML **Class Diagram** in Appendix C (page 8) shows the relationship between FOUR classes and snippet code of some of the methods implementation. Both `SavingAccount` and `InvestmentAccount` classes have a reference to the `DataPresenter` object in order to know the type of format (XML or HTML). `DataPresenter` class also has a reference each to the `SavingAccount` and `InvestmentAccount` objects in order to read their content and present the format accordingly.

Note: Question No. 4 continues on Page 5

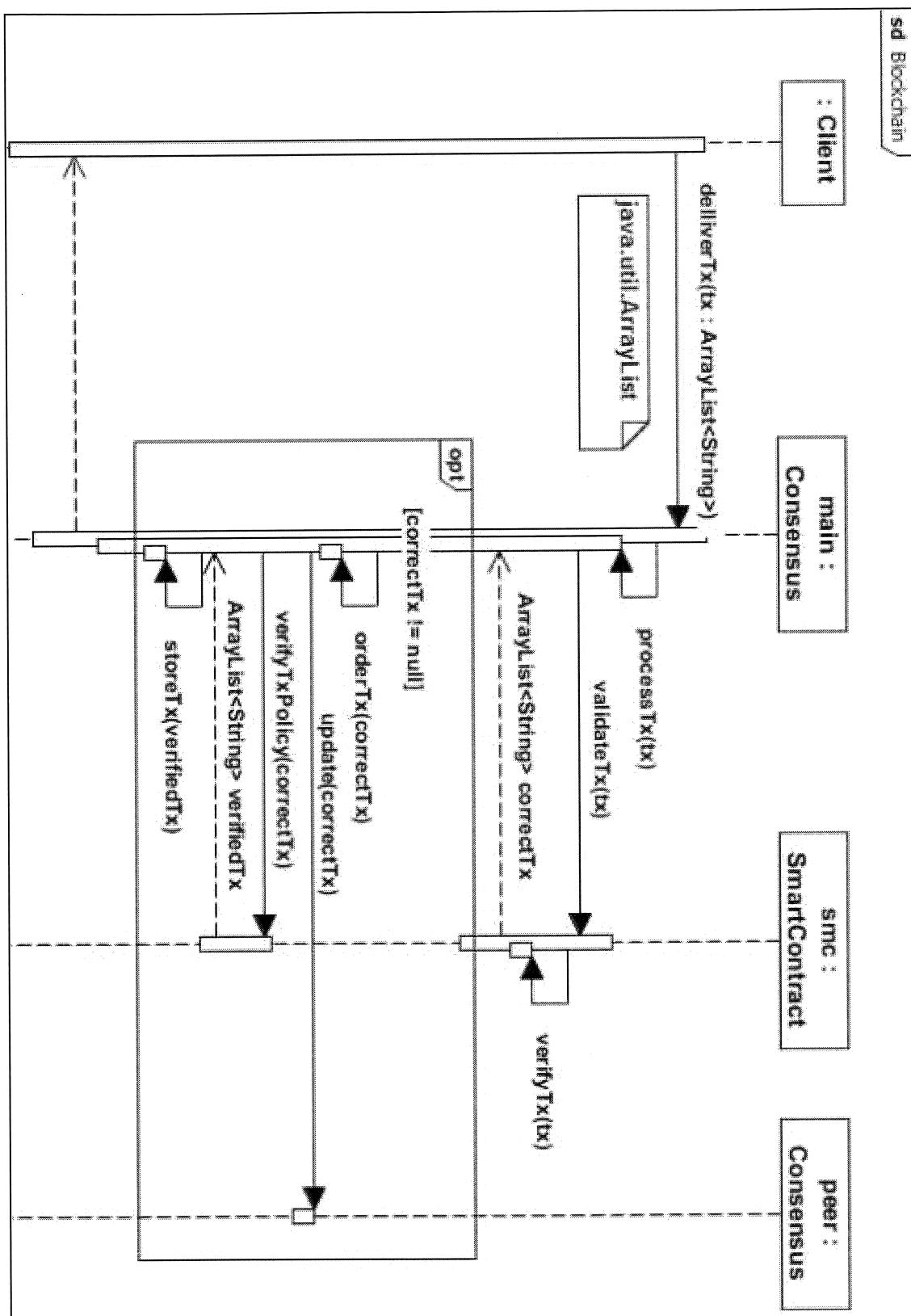
It is suggested that a new RetirementAccount class will be added and the data should be presented also in JSON format.

- (i) State and explain ONE design issue of the current design.  
(3 marks)
- (ii) Suggest and explain, with a Class Diagram, how you can improve the current design to cater to the new requirements 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.***  
(7 marks)

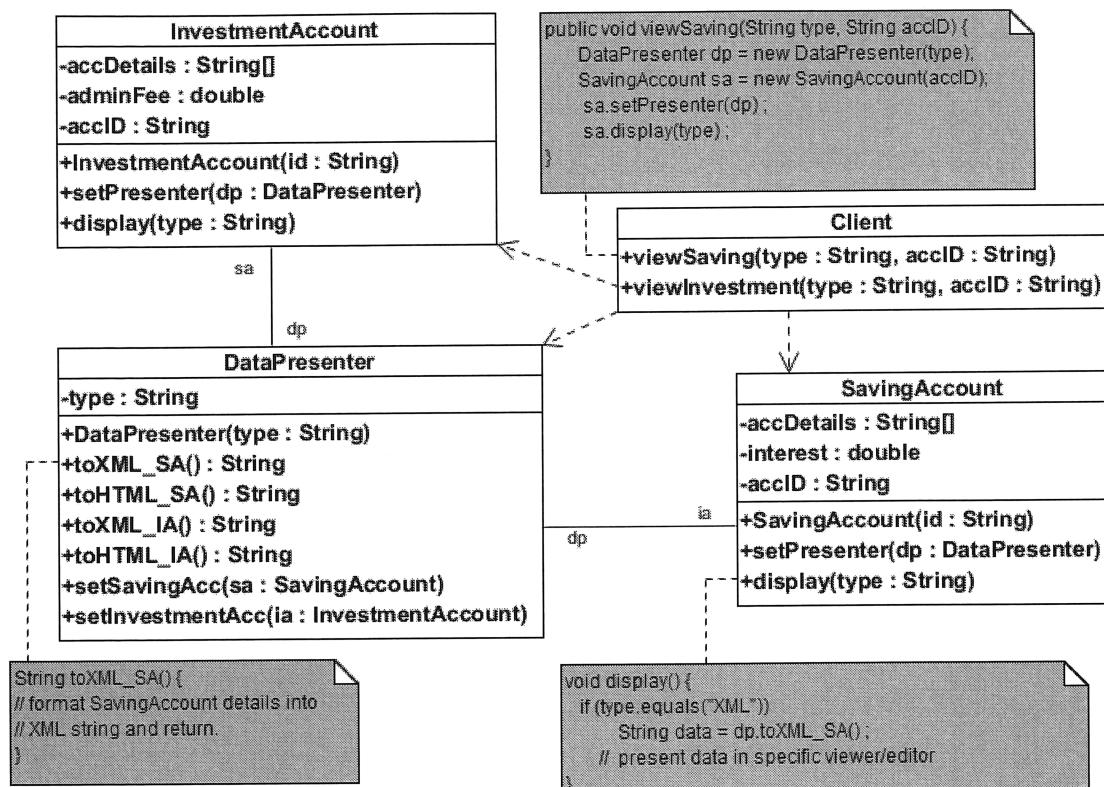
## APPENDIX A: Class Diagram



## APPENDIX B: Sequence Diagram



## APPENDIX C: Class Diagram



END OF PAPER







**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.