

**NANYANG TECHNOLOGICAL UNIVERSITY**

**SEMESTER 1 EXAMINATION 2018-2019**

**CE2002/CZ2002 – OBJECT-ORIENTED DESIGN & PROGRAMMING**

Nov/Dec 2018

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 2.
6. Appendix B shows the Class Diagram referenced by Question 3.
7. Appendix C shows the Sequence Diagram referenced by Question 3.

- 
1. (a) (i) Which object-oriented feature can help protect what you have and protect your data? Briefly explain this feature. With the help of a simple piece of Java code, explain how to realize this feature in Java programming.  
(6 marks)
  - (ii) Due to the feature in Q1(a)(i), which parts of a class are visible to and can be used by external classes? What is the specific term/name for these parts?  
(4 marks)
  - (b) Write Java code to demonstrate how to define a *static* variable. And, list THREE differences between a *static* variable and an *instance* variable.  
(5 marks)

Note: Question No. 1 continues on Page 2

- (c) What is *object composition*? Which relationship in OO does it refer to? Write a simple piece of Java code to demonstrate it. (5 marks)
- (d) Explain what *Polymorphism* is in object oriented programming, and list THREE concepts that are closely related to Polymorphism. (5 marks)
2. Study the UML **Class Diagram** in Appendix A (page 6). Assume that `ClassA` has already been completely implemented. The `record()` methods in `ClassA` keep a record of the parameter value in a database.
- (a) Write Java code for `InterfaceA`, `InterfaceB`, and `ClassB`. Note that if a `record()` method has more than one parameter that is of the `String` type, it simply keeps a record of the concatenation of the parameters. If a `record()` method has more than one parameter that is of the `int` type, it simply keeps a record of the sum of the parameters. (12 marks)
- (b) Add a concrete class named `ClassC` that extends `ClassB`. Draw an UML class diagram and write Java code for this class. (6 marks)
- (c) EXPLAIN the outcome, the class's method called or used (if any) and the type of casting for **each line** of the following codes during **compile-time** and **runtime**.
- (i) 

```
InterfaceA a = new ClassB() ;
ClassA b = (ClassA) a ;
b.record("1", "2") ;
```

 (3 marks)
- (ii) 

```
InterfaceB a = new ClassC() ;
ClassB b = (ClassB) a ;
b.record("1") ;
b.record(1, 2, 3) ;
```

 (4 marks)

3. (a) The UML **Class Diagram** in Appendix B (page 7) shows the relationships of ONE interface : `IFigure3D` and THREE classes : an abstract class `Figure2D`, a concrete class `Circle`, and a concrete class `Cone`. Study the class diagram and the details depicted carefully.

Additional details relevant to this question are provided below.

The parameters of the `Cone` class constructor, i.e., `name`, `ht`, `s_ht` and `rad` are used to initialize the `name`, `height`, `slanted_ht` and instantiate the `Circle` object respectively. The `print()` function prints the details of the figure by first using `Figure2D`'s `print()` function, followed by `Circle`'s `print()` function and finally, prints its height and slanted height in this format :

```
[ Figure2D print() output ]
[ Circle print() output ]
Cone height: <height>, slanted height: <slanted_ht>
```

Example :

```
Figure : My 2D Figure
Circle radius: 1.5
Cone height: 3.5, slanted height: 2.5
```

The area and volume of a `Cone` are given as  $\pi r l + \pi r^2$  and  $\frac{1}{3} \pi r^2 h$ , where  $r$  is the radius,  $l$  is the slanted height and  $h$  is the height. The `getPI()` method of the `Circle` class will return the value of  $\pi$ .

- (i) Write the C++ code for the `IFigure3D` interface in a header file, **figure3d.h**, as depicted in the class diagram. (2 marks)
- (ii) Write the C++ code for the `Cone` class completely with its declaration and implementation in the file, **cone.cpp**. You can assume that the `Figure2D` class and `Circle` class are already implemented in the **figure2d.h** and **circle.h** header files respectively.

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

(11 marks)

Note: Question No. 3 continues on Page 4

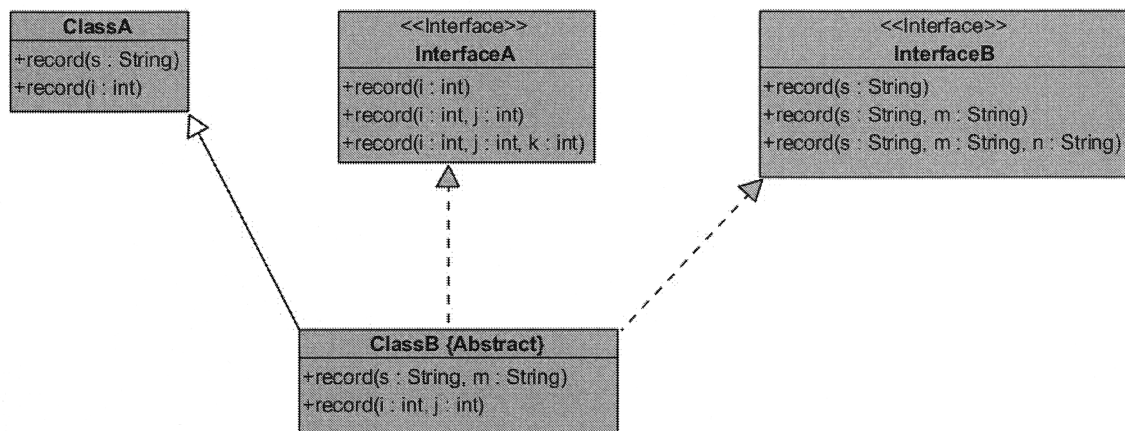
- (b) The UML **Sequence Diagram** in Appendix C (page 8) shows the objects' interaction of a scenario flow in a particular application. Using the details depicted in the diagram, write the preliminary JAVA code for the `Dealer` and `Player` class and their methods. You may make appropriate assumptions on the method parameters, return types and return value(s) if they are not stated in the diagram. (12 marks)
4. (a) OVRS is an online video rental service which offers hit Hong Kong and Korean television (TV) series. It allows customers to rent TV series online for a period of time depending on the length of the series. The TV series can be either a Hong Kong or Korean drama. Each series has a title, a short description, the total number of episodes, a list of regular cast and a rental price. Each episode has an episode number, the date it was first aired and the length of the episode in minutes. The cast consists of the main actors/actresses in the series and will have their name, gender and nationality. In order to rent the TV series, a user has to register as a customer by providing his/her name, userID, password, contact number, email address and the preferred mode of payment. The current supported mode of payment is either PayPal or credit card. In future, it will include other modes of payment like Alipay and even virtual currency. Each rental transaction will record the customer, the series, start date and end date. Each series can be concurrently be rented by multiple customers.
- 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. You need not show the class methods but include interface and enumeration if appropriate. (15 marks)
- (b) Design principles are applied so as to manage the impact of change to the software application. However, in most software lifecycle, change is inevitable and there is bound to have changes.

Note: Question No. 4 continues on Page 5

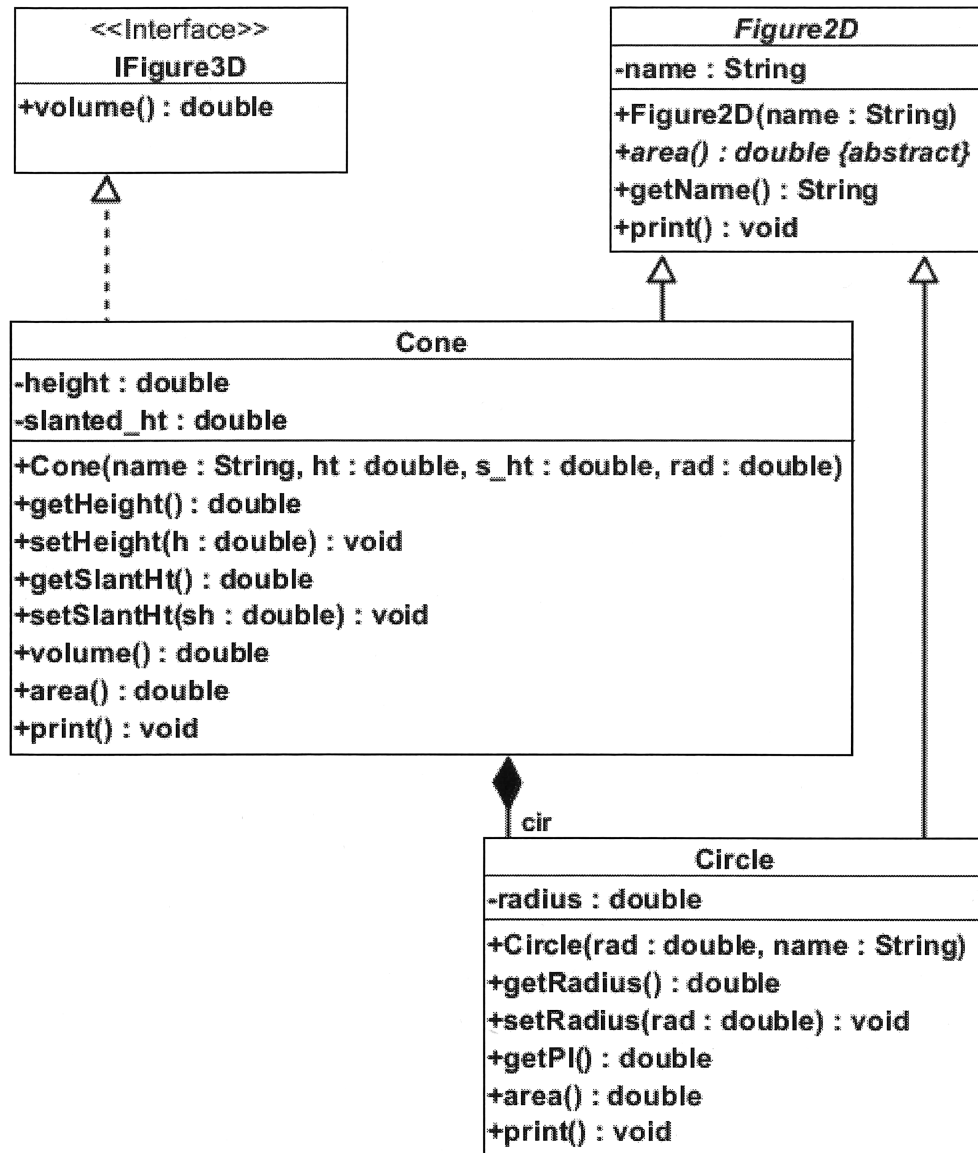
CE2002/CZ2002

- (i) Describe TWO cases where we need to make changes to software, and explain how a good design can manage the impact of change.  
(4 marks)
- (ii) Use any TWO of the SOLID design principles to explain how they help in managing the impact of change.  
(6 marks)

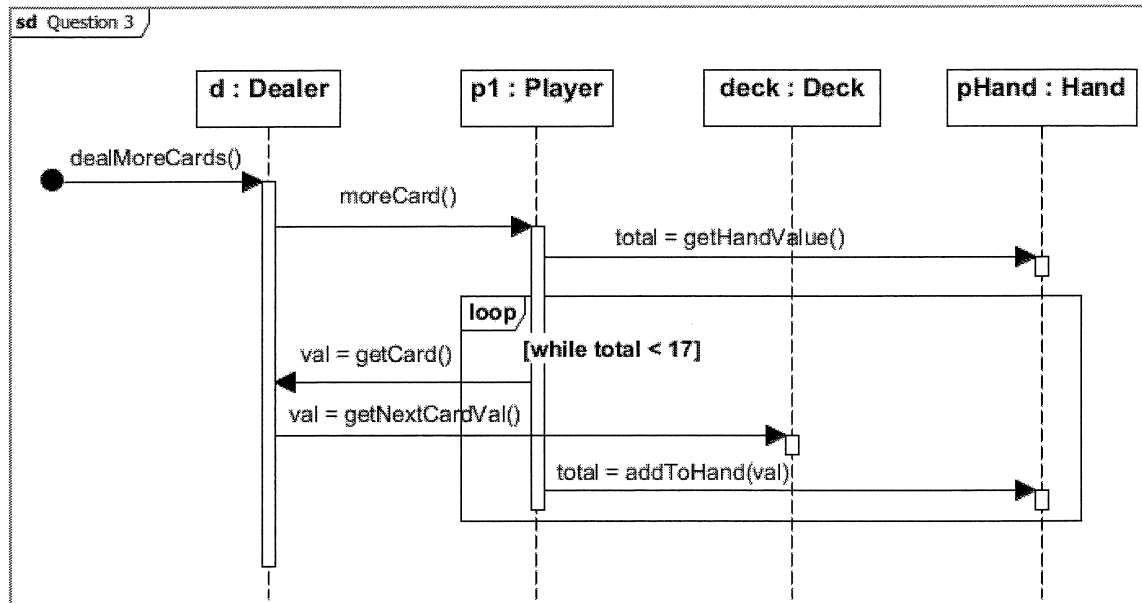
## APPENDIX A : Class Diagram



## APPENDIX B : Class Diagram



## APPENDIX C : Sequence Diagram



END OF PAPER









**CE2002 OBJECT ORIENTED DESIGN & 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.