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

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2015-2016

CE2002/CZ2002 – OBJECT ORIENTED DESIGN & PROGRAMMING

Nov/Dec 2015 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 Java code listing referenced by Question 1.
6. APPENDIX B shows the Class Diagram referenced by Question 3.
7. APPENDIX C shows the Sequence Diagram referenced by Question 4.

---

1. (a) Explain what is *object composition*. Demonstrate object composition by defining example classes in Java and creating objects using the defined example classes.

(5 marks)

(b) List **THREE** ways of method overriding.

(6 marks)

Note: Question No. 1 continues on Page 2

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

(c) Study the class hierarchy diagram in APPENDIX A (page 7).

(i) In such an inheritance hierarchy of classes, explain how Java resolves method calls when a message is sent to an object.

(4 marks)

(ii) Assume that `ClassF objectB = new ClassF();`

Identify the class, of which the `print(...)` method will be invoked for each of the statements below. If there is an error, explain why the error occurs.

```
objectB.print("OODP");
objectB.print(2002, "OODP");
objectB.print("HELLO", "OODP");
objectB.print(2002);
objectB.print(2002, 2015);
```

(10 marks)

2. You are required to write a Java application program to manage bank accounts. Each account has the information about the name of the account owner, account number, and the amount of money in the account. You can deposit money into the account and also withdraw money from the account. There are two kinds of accounts. One is the normal account where the withdraw amount cannot exceed the amount available in the account. Another is the privileged account where the withdraw amount can exceed the amount available in the account, but up to a certain limit. The program finally prints out the available amount in the account.

(a) Write the code for the **abstract** class `Account` that has the instance variables of `name`, `accountNo` and `amount`, a constructor, the instance methods of `getAmount` and `deposit`, and an abstract method `withdraw`.

- `name`: name of the account owner.
- `accountNo`: account number.
- `amount`: amount of money available in the account.
- the constructor initializes the values of all instance variables.
- `getAmount`: accessor method to get the amount of money available in the account.

Note: Question No. 2 continues on Page 3

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

- `deposit`: an instance method to deposit a certain amount of money into the account.
- `withdraw`: an abstract method. This method has the parameter which is the amount of money to be withdrawn from the account.

(8 marks)

(b) Write the code for the subclass `NormalAccount` derived from the `Account` class. It has a constructor to initialize the values of the inherited instance variables. It also has a `withdraw` method and provides implementation for the method. The `withdraw` method basically withdraws a certain amount of money from the normal account. If the amount to be withdrawn from the account exceeds the available money in the account, it will display a warning message "Over Limit!".

(5 marks)

(c) Write the code for another subclass `PrivilegedAccount` derived from the `Account` class. It has an instance variable `limit`. This instance variable determines up to how much the withdrawn amount can exceed the available amount of money in the account.

The `PrivilegedAccount` class has a constructor to initialize the values of the inherited instance variables and the instance variable `limit`. It also has a `withdraw` method and provides implementation for the method. The `withdraw` method basically withdraws a certain amount of money from the privileged account. If the amount to be withdrawn from the account exceeds the available money in the account with the amount which is larger than `limit`, it will display a warning message "Over Limit!".

(5 marks)

(d) Write an application class `AccountApp` to have two `withdraw` methods. One `withdraw` method withdraws a certain amount of money from a normal account, and another `withdraw` method withdraws a certain amount of money from a privileged account. Also write the `main` method in the application class to demonstrate **static binding** of the `withdraw` methods for a normal account and a privileged account.

Note: Question No. 2 continues on Page 4

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

To demonstrate **dynamic binding** of the `withdraw` method, write another version of the application class `AccountApp` to have only one `withdraw` method in the application class, regardless of which `Account` subclass passed as argument of the `withdraw` method.

(7 marks)

3. (a) The UML **Class Diagram** in Appendix B (page 8) shows the class relationships of FOUR classes: an *abstract* class `Weapon`, a `Gun` class, a `Sword` class and a `Player` class. Study the class diagram carefully. Additional details relevant to this question are provided below.

The parameters of the `Weapon` class constructor `a`, `w` and `s` initialize the `attackRate`, `weight` and `sound` respectively. The `use` method in the `Weapon` class prints the `sound` of the weapon to the console.

The parameters of the `Gun` class constructor `a`, `w`, `s`, `p`, `r` and `am` initialize the `attackRate`, `weight`, `sound`, `power`, `range` and `ammo` respectively. The `loadAmmo` method adds the parameter, `am`, to the `ammo` value and returns the total `ammo` value. The `use` method overrides the `use` method in the `Weapon` class by decrementing the `ammo` value by ONE, if the `ammo` value is not ZERO, and also calls the `use` method in the Weapon class. The `calDamage` method implements the `calDamage` *abstract* method in the `Weapon` class by returning the result of the multiplication of the `attackRate` and `power`.

(i) Write the C++ code for the `Weapon` class.

(6 marks)

(ii) Write the C++ code for the `Gun` class. You should include all necessary error checking in your code.

(7 marks)

Note: Question No. 3 continues on Page 5

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

(iii) Write a C++ *non-member* function which overloads the "+" operation to assign a `Weapon` object to the right hand of a `Player` object. For example, by doing the following:

```
player1 = player1 + gun,
```

a gun will be assigned to player1's right hand (right operand).

(4 marks)

(b) With your understanding of the Design Principles, answer the following questions:

(i) Explain *Role versus Inheritance* and give an example.

(4 marks)

(ii) Explain *Liskov Substitution Principle (LSP)* and by using the example of Circle and Ellipse, explain what is meant by violation of LSP.
[*Hint: for an ellipse with major axis of length A (the longer radius) and minor axis of length B (the shorter radius), the area is* $\pi \times A \times B$].

(4 marks)

4. (a) Study the following description of a Staff Project Management System (SPMS):

SPMS is a system that maintains a centralized repository of all staff information, the project they are assigned and the tracking of project issues. SPMS allows department managers to easily access the relevant information on staff (including managers) and projects as well as tracking a project's progress.

SPMS stores the staff details like staff identification number, name, the date of joining, department, the assigned projects and staff's skillsets. Each staff belongs to a department which is managed by a manager and supported by a secretary. There are also supervisors in the department to supervise at most FOUR staff. Staff will be assigned to ONE to TWO projects depending on their skillsets. Skillsets like certification name, the level, awarded date and date of expiry are stored.

Note: Question No. 4 continues on Page 6

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

A project is funded externally by a company with a budget to last the duration of the project. Each project can have multiple sub-projects depending on its complexity. A staff will be assigned as a project leader to each project, either main or sub. Project issues are tracked by assigning an identification number and providing details like title, detailed description, status, reported date and last updated date.

You are tasked to identify the entity classes needed to build the SPMS 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), multiplicities, role names and association names, if any. ***You need not show the class methods***.

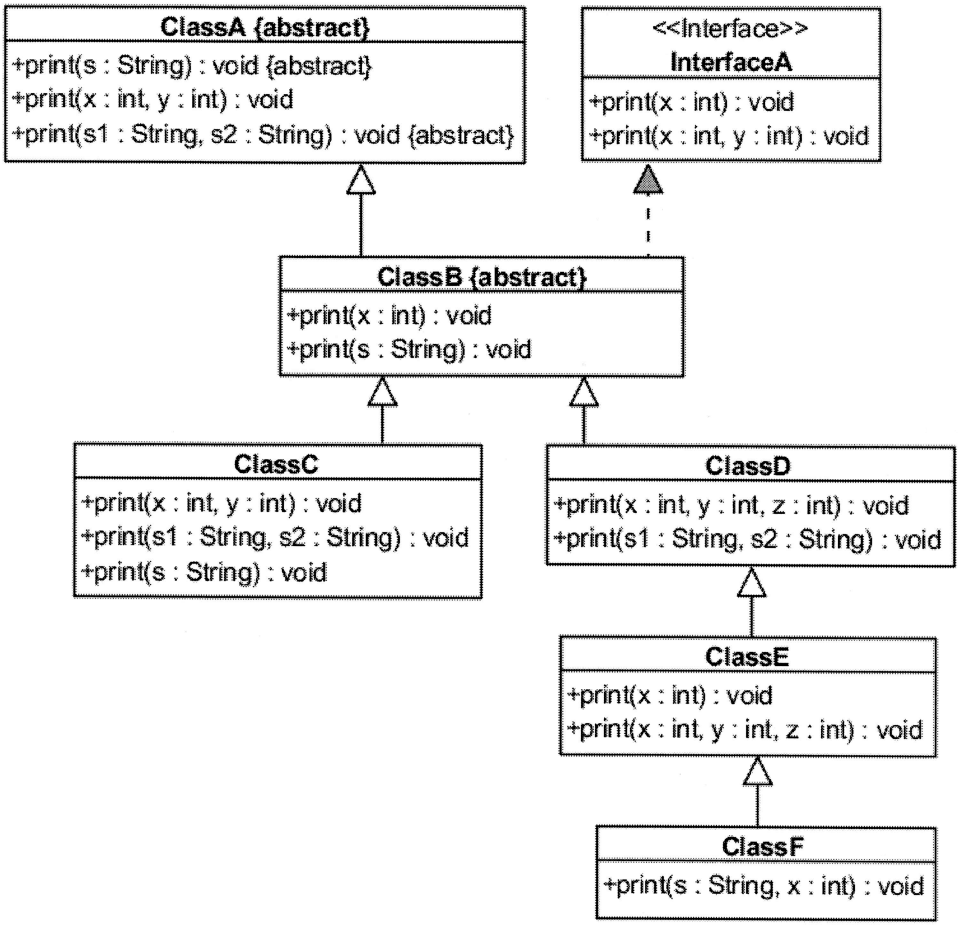
(15 marks)

(b) The UML **Sequence Diagram** in Appendix C (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 and methods for the class `HandlerA`. You may make appropriate assumptions on the method parameters, return types and return value(s) if they are not stated in the diagram.

(10 marks)

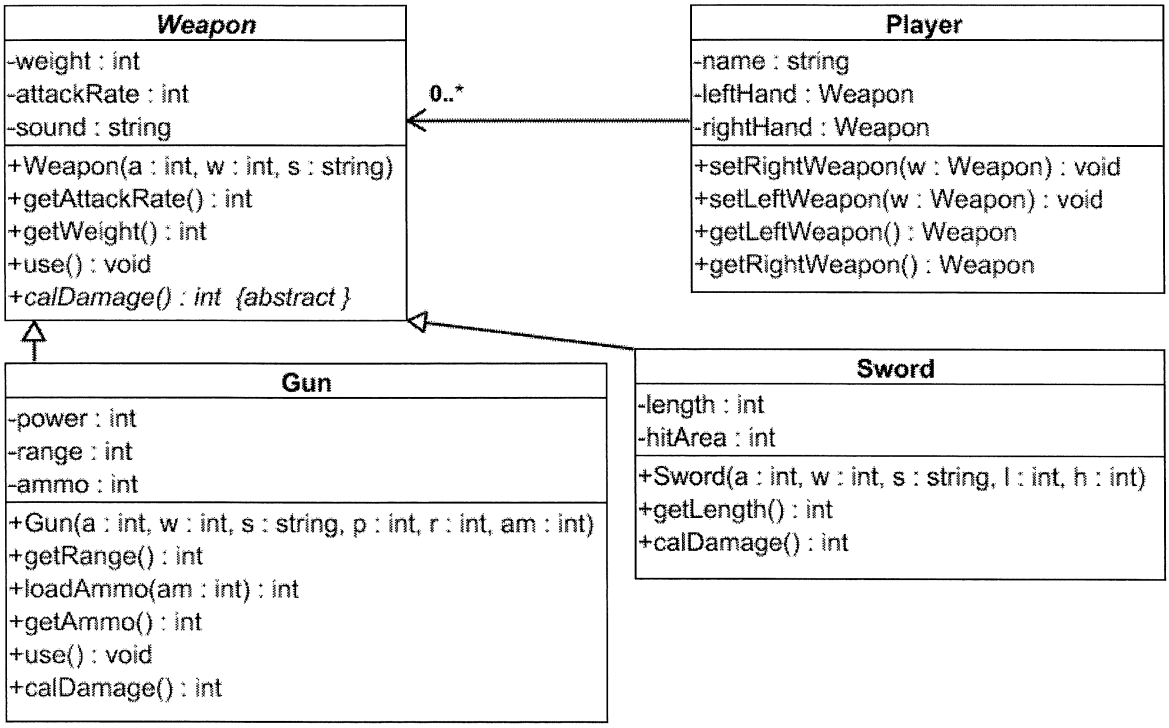
ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

APPENDIX A : Class Hierarchy



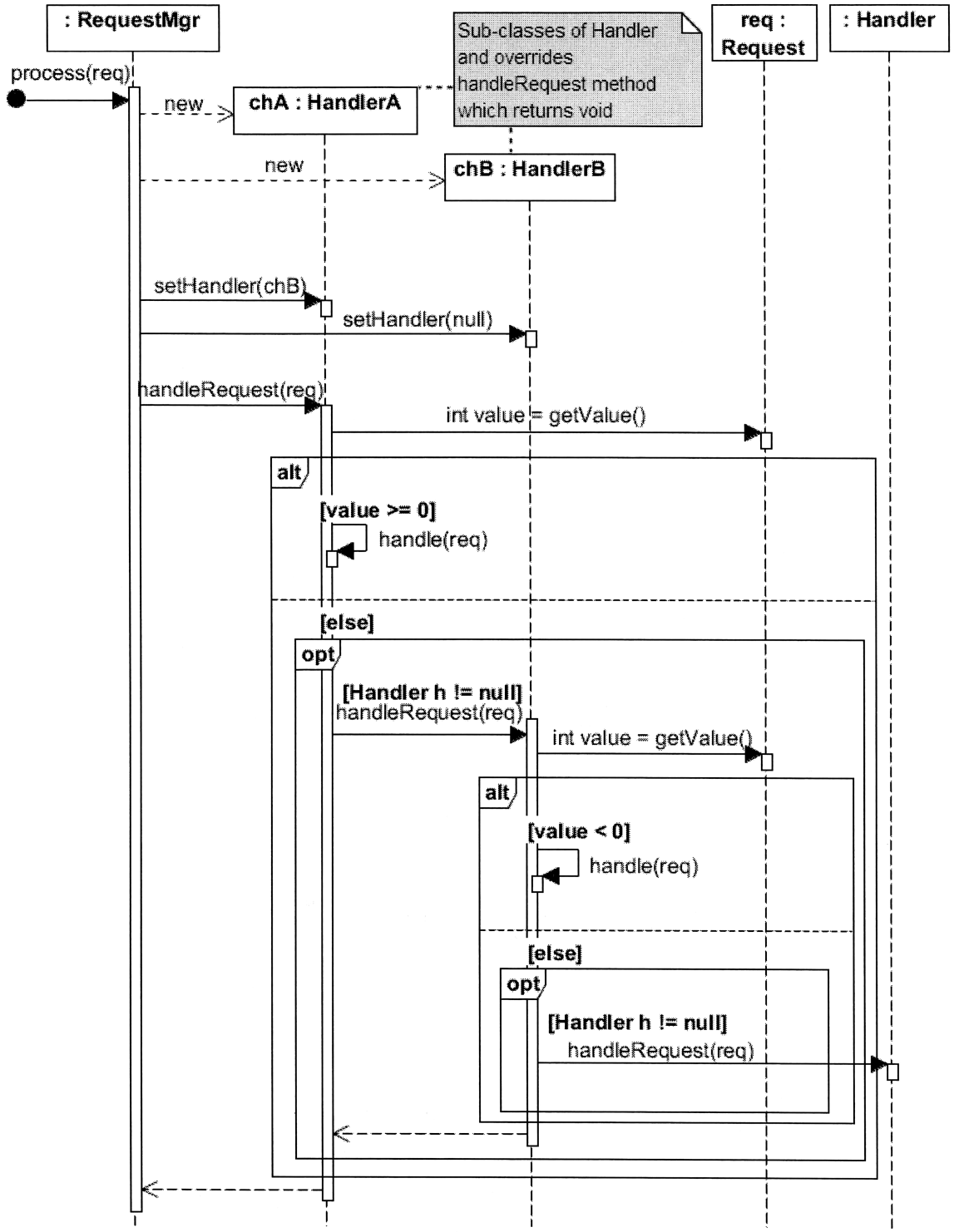
ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

APPENDIX B : Class Diagram



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

APPENDIX C : 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

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.