

NANYANG TECHNOLOGICAL UNIVERSITY**SEMESTER 2 EXAMINATION 2020-2021****CE2002/CZ2002 – OBJECT ORIENTED DESIGN AND PROGRAMMING**

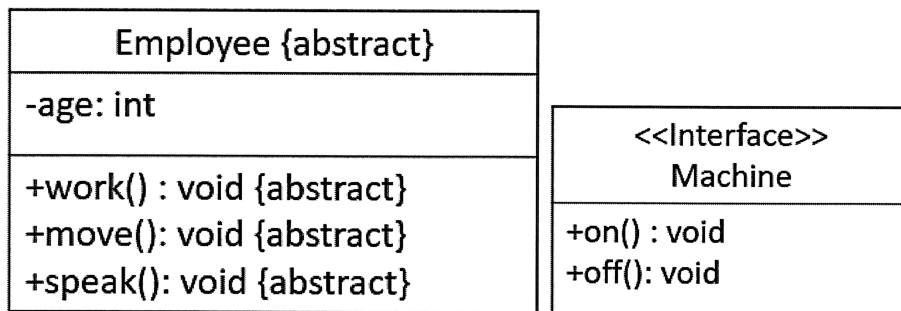
Apr/May 2021

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 the Sequence Diagram referenced by Question 3(b).

1. (a) Study the partial UML class diagram in Figure Q1.

**Figure Q1**

The methods simply display the method name and do not return any value. For example:

- The method speak() just displays the string “speak”.

Note: Question No. 1 continues on Page 2

- (i) Draw the COMPLETE class diagram by adding the following CLASSES/INTERFACES (with their necessary attributes and/or methods) into the partial diagram in Figure Q1:
- Class Cleaner extends Class Employee, It provides the implementation for rest (), and speak () methods. It does not contain any other method.
 - Interface Computer extends Interface Machine, It defines the exeProgram () method. It does not contain any other method.
 - Class RobotCleaner extends Class Cleaner, and implements Interface Computer. It provides implementations of all necessary methods to make it a concrete class.
- (5 marks)

- (ii) Write the Java code for all the CLASSES/INTERFACES in the complete class diagram and their methods.
- (8 marks)

- (b) Write out the output of the following program and briefly explain the reason.

```
class A
{
    public A ()
    {
        System.out.println("construcotr A");
    }

    void method(int a)
    {
        System.out.println("A1");
    }

    void method(double a)
    {
        System.out.println("A2");
    }
}
```

Note: Question No. 1 continues on Page 3

```
class B extends A
{
    public B ()
    {
        System.out.println("construcotr B");
    }

    void method(double a)
    {
        System.out.println("B1");
    }
}

class C extends B
{
    public C ()
    {
        System.out.println("construcotr C");
    }

    void method(int a)
    {
        System.out.println("C1");
    }
}

public class Test
{
    public static void main(String[] args)
    {
        B ob = new B();
        C oc = new C();
        ob.method(100);
        oc.method(100);
        ob.method(9.9);
        oc.method(9.9);
    }
}
```

(12 marks)

2. (a) According to the following partial code, instructions in comment statements, and the output of sample run as follows:

```
class TestClass
{
    private double price;
    private String name;
    static int numberOfProduct = 0;

    // definition of TestClass constructor

    public static void greetings(){
        // definition of greetings from TestClass
    }
    public void update(double price, String name){
        // definition of update
    }
    public void print(){
        // definition of print
    }
    public void printNumberOfProduct(){
        // definition of printNumberOfProduct
    }
}

public class TestClassApp
{
    public static void main(String[] args){
        //print the greetings from TestClass
        //construct a TestClass object called myProduct
            //and initialize attribute name
        //print values of myProduct's attributes
        //update myProduct
        //print values of myProduct's attributes
        //print myProduct's numberOfProduct
        //construct a TestClass object called tomProduct
            and initialize attribute name
        //print tomProduct's numberOfProduct
    }
}
```

Sample run:

```
Hello, welcome to our product line.
Amazing: costs S$0.0
AmazingII: costs S$999.99
number = 1
number = 2
```

- (i) Write the complete Java program for class TestClass.

(9 marks)

Note: Question No. 2 continues on Page 5

(ii) Write the complete Java program for class TestClassApp.

(9 marks)

(b) Write out the output of the following program and explain the reason in details.

```
class X
{
    void compare(int a, int b)
    {
        if (a == b) System.out.println("equal in X");
        else System.out.println("different in X");
    }
}

class Y extends X
{
    void compare(int a, int b)
    {
        if (a == b) System.out.println("equal in Y");
        else System.out.println("different in Y");
    }
}

class Z extends Y
{
    void compare(int a, int b)
    {
        if (a == b) System.out.println("equal in Z");
        else System.out.println("different in Z");
    }
}

public class TestQ
{
    public static void main(String[] args)
    {
        X x1 = new Z();           //line 1
        x1.compare(10, 10);       //line 2
        Y y1 = (Y)x1;            //line 3
        y1.compare(10, 20);       //line 4
    }
}
```

(7 marks)

3. (a) Study the following description of an application :

MOVPUR is an application to computerize the processes of making online purchase of movie tickets. It will be used by the movie-goers and cinema staff in each cineplex in different locations. Each cineplex will have 3 or more cinema theatres. The movie ticket price is charged according to the age group of movie-goer, day of the week or public holiday. The age group can be adult, senior citizen or child. MOVPUR allows movie-goers to query the movie listings and show times. From the movie listings, movie-goer can also view the information about the movie like title, showing status, classification, synopsis, directors, casts and reviewers' ratings. The showing status can be "Coming Soon", "Preview" or "Now Showing". The movie classification can be PG, NC16, M18 or R21. Only registered movie-goer with a valid identification and password can make online purchases. Each movie ticket will indicate the movie title, show time and date, seat number and the cinema. Registered movie-goer can also check their history of bookings.

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 relationships between classes, enumeration, relevant attributes (at least TWO), logical multiplicities, meaningful role names, association names and constraint(s), if any. You need not show the class methods.

(13 marks)

- (b) The UML **Sequence Diagram** in Appendix A (page 9) shows the objects' interactions of a scenario flow in a particular application. Using the details depicted in the diagram, write the preliminary Java code for the `Work` class 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) You are required to write a C++ 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. The account owner can deposit money into the account and also withdraw money from the account.

Note: Question No. 4 continues on Page 7

- (i) Write the C++ code, in an `account.h` header file, for the class `Account` that has the private instance variables of `name`, `accountNo` and `amount`, a constructor, the public instance function of `getAmount` and `deposit`, and an abstract public function `withdraw` with the following details:
- `name : String` : name of the account owner.
 - `accountNo : String` : account number.
 - `amount : float` : amount of money available in the account.
 - the constructor initializes the values of all instance variables.
 - `getAmount()` : accessor function to get the amount of money available in the account.
 - `deposit(amount : float)` : an instance function to deposit an amount of money into the account by adding to the current balance amount.
 - `withdraw(amount : float)` : an instance function to withdraw an amount of money from the account by subtracting from the current balance amount. But if the amount to be withdrawn from the account exceeds the available money in the account, it will display a warning message "Over Limit!".

(10 marks)

- (ii) Write the C++ code for the subclass `PrivilegedAccount` derived from the `Account` class. The subclass has a constructor to initialize the values of the inherited instance variables and its instance variable `limit` (`float`). It also has a `withdraw` function and provides the implementation for the function. The `withdraw` function overrides the `Account` class `withdraw` by basically withdrawing an amount of money from the privileged account by subtracting from the current balance amount. But if the amount to be withdrawn from the account exceeds the available money in the account with an amount which is larger than `limit`, it will display a warning message "Over Limit!"

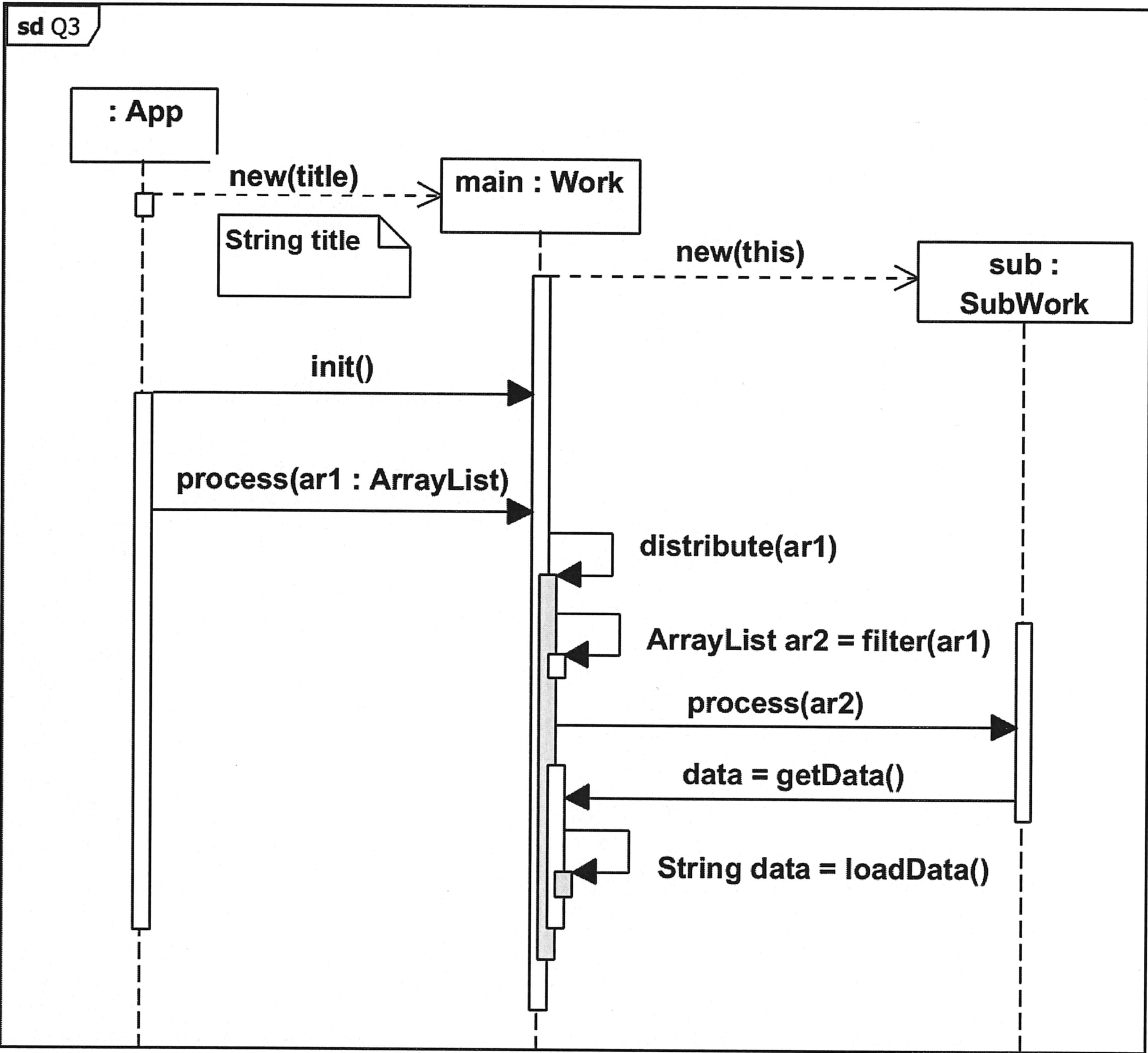
(5 marks)

- (b) Further enhancements were suggested to the design depicted in Q4(a) by including the recording of all deposit and withdraw transactions in `Account` class and its subclasses, and, allowing these transactions to be formatted in different formats like XML (eXtensible Markup Language) or JSON (JavaScript Object Notation).

Note: Question No. 4 continues on Page 8

- (i) By applying the SOLID design principles, suggest and explain using a Class Diagram how you would design to incorporate the TWO suggested enhancements and minimize the impact of the change/s to the existing classes at the same time.
(6 marks)
- (ii) Use TWO of the SOLID design principles to explain how they were applied.
(4 marks)

APPENDIX A: 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.