NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2018-2019

CE2002/CZ2002 – OBJECT ORIENTED DESIGN & PROGRAMMING

Apr/May 2019 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).
6. APPENDIX B shows the Class Diagram referenced by Question 4(a).

---

1. (a) Study the partial UML class diagram in Figure Q1a.

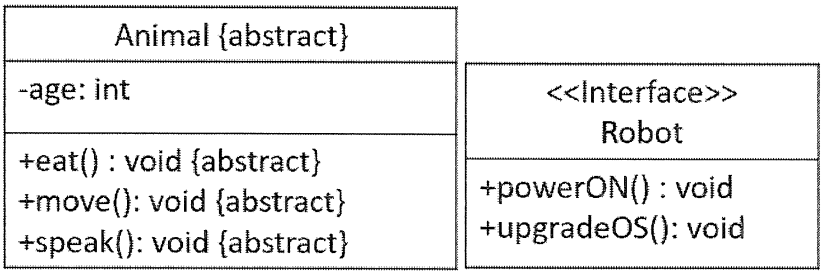


Figure Q1a

(i) Draw the COMPLETE class diagram by adding the following CLASSES (with their necessary attributes and/or methods) into the partial diagram in Figure Q1a.

Note: Question No. 1 continues on Page 2

- Class **Cat** extends Class **Animal**, it provides implementation for the `eat()`, `move()`, and `speak()` methods. It does not contain any other method.

- Class **RobotCat** extends Class **Cat**, and implements Interface **Robot**. It provides implementation for the `powerOn()` method and `eat()` method. It does not contain any other method.

- Class **RobotCatAI** extends Class **RobotCat**, and provides implementation for `upgradeOS()` method and `imageDetection()` method. It does not contain any other method

(6 marks)

(ii) Write the Java code for all the CLASSES in the complete class diagram and their methods. All the methods simply display the method name and do not return any value. Fox example, the method `eat()` just display string eat.

(9 marks)

(b) Consider the following Java classes. For each line, from line 1 to line 4, state the output and briefly explain the reason.

```
package p1;
public class A {
     public void fun (double d)
     { System.out.println("AD"); }
     void fun(String tr)
     { System.out.println("AS");}
}

package p1;
public class B extends A {
     public void fun (Object o)
     { System.out.println("BO"); }
}

package p1.s1;
import p1.*;
class D extends A {
     public void fun (Object o)
     { System.out.println("DO"); }
}
```

Note: Question No. 1 continues on Page 3

```
public class Test {

     public static void main(String[] args) {
          B b = new B();
          D d = new D();

          b.fun(3.14);   //line 1
          d.fun(3.14);   //line 2
          b.fun("test"); //line 3
          d.fun("test"); //line 4
     }
}
```

(10 marks)

2. (a) Consider the following Java classes. For each line, from line 1 to line 5, state the output and briefly explain the reason.

```
class A {
          public void fun (double d)
          { System.out.println("A"); }
}

class B {
          public void fun (int i)
          { System.out.println("B"); }
}

class C extends A {
          public void fun (int i)
          { System.out.println("C"); }
}

class D extends B {
          public void fun (double d)
          { System.out.println("D"); }
}

class E extends B {
          public void fun (double d)
          { System.out.println("E"); }
          public void fun (int i)
          { System.out.println("F"); }
}
```

Note: Question No. 2 continues on Page 4

```
public class Test {
          public static void main(String[] args){
               C c = new C(); c.fun(6); //line 1
               D d = new D(); d.fun(6); //line 2
               A x = new C(); x.fun(6); //line 3
               B y = new D(); y.fun(6); //line 4
               B z = new E(); z.fun(6); //line 5
          }
}
```

(10 marks)

(b) Consider the following Java classes. For each line, from line 1 to line 5, state the output and briefly explain the reason.

```
class AA {
    int a = 10;
    static int b = 0;

    AA(){print(); }

    AA(int a){
          this.a = a;
          print();
          b++;
    }

    void print() {
        System.out.println(a);
        System.out.println(b);
    }
}
class A1 extends AA {
    A1(){ System.out.println("A1");}
}
class A2 extends AA {
    A2(int a){
        super(a);
        System.out.println("A2");
     }

    void print() {
        System.out.println("A2 print");
        System.out.println(super.b);
    }
}
```

Note: Question No. 2 continues on Page 5

```
public class Test2 {
        public static void main(String[] args)
        {
            AA a = new AA();        //line 1
            A1 a1 = new A1();       //line 2
            A2 a2 = new A2(4);      //line 3
            a2.print();             //line 4
            a1.print();             //line 5
        }
}
```

(15 marks)

3. (a) Discount Travel is an online service provider that provides the public with a list of holiday trips that have not been sold by travel agencies. In order for a customer to purchase the holiday trips, the customer needs to register by providing his/her name, email, contact number and credit card details. A holiday consists of trips to different destinations. A destination has a name, description and it can be a place-of-interest, famous eatery, wholesale factory or others which can be included in the future. Each trip takes the customer to a single country and state where he/she will tour the different destinations. Different trips are proposed at discounted prices and include a description of what each covers. A holiday will provide a description of the various trips, a reference number, the departure and arrival dates, the maximum number of participants and the overall price. A customer will select the holidays he/she wants to purchase and the number of participants by providing the participants' names and passport numbers. Multiple holidays can be selected *as long as the dates do not coincide*. A holiday is only confirmed if an order is placed by the customer by making payment using his/her credit card. The order will list each of the holiday items, number of participants, a reference number and the price. An invoice will be printed with the order and the payment details for the customer's reference.

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, 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)

Note: Question No. 3 continues on Page 6

(b) The UML **Sequence Diagram** in Appendix A (page 8) 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 `Runner` 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) The UML **Class Diagram** in Appendix B (page 9) shows the relationships of TWO interfaces : `IPlottable` and `IDrawable` together with THREE classes : `ScatterPlot`, `Data`, and `Axes`. Study the class diagram and the details depicted carefully.

Additional details relevant to this question are provided below.

The parameters of the `Scatter` class constructor `desc`, `title`, x, y and `d` are used to initialize its attribute `desc`, instantiate the `Axes` object and the `Data` object respectively. The `plot()` function simply prints `"plotting..."` and the `draw()` function calls the `draw()` function of the `Axes` class before printing `"drawing..."`.

(i) Write the C++ code for the `IDrawable` interface in a header file, **IDrawable.h**.

(2 marks)

(ii) Write the C++ code for the `ScatterPlot` class with all declarations and the constructor implementation in a header file, **scatterplot.h** and the implementation of the `plot()` and `draw()` functions in an implementation file, **scatterplot.cpp**. You can assume that the `Axes` class and `Data` class are already implemented in the **Data.h** header file.

[You should use the appropriate C++ keyword/s and pointers, if necessary, to ensure the code will execute as expected.]

(13 marks)

(b) Further enhancements were suggested to the design depicted in Q4(a) to include a third axis, z, in addition to the current 2 axes (x,y) and also to allow the axes to be rotated, *including with the current 2 axes.*

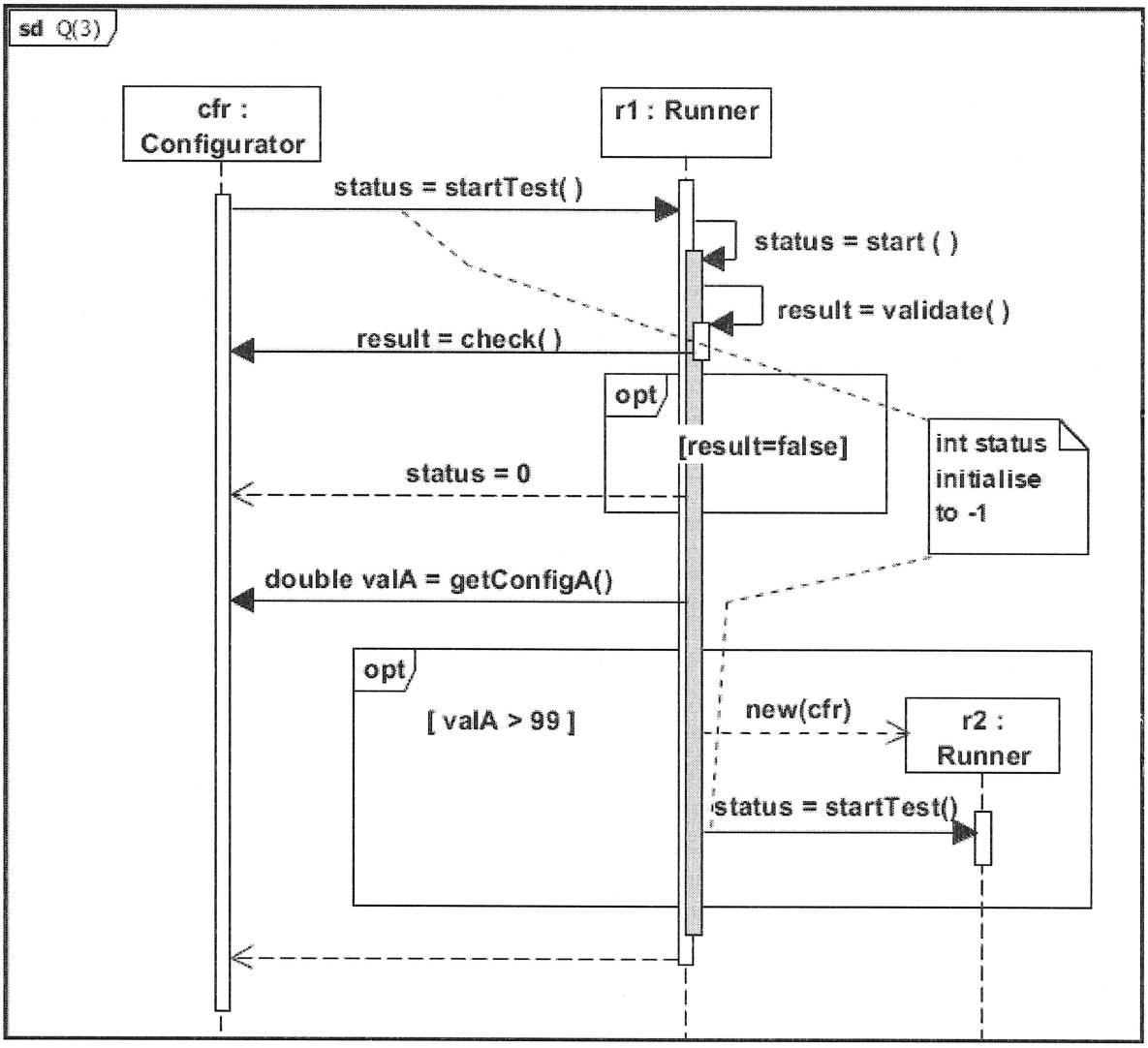
Note: Question No. 4 continues on Page 7

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.

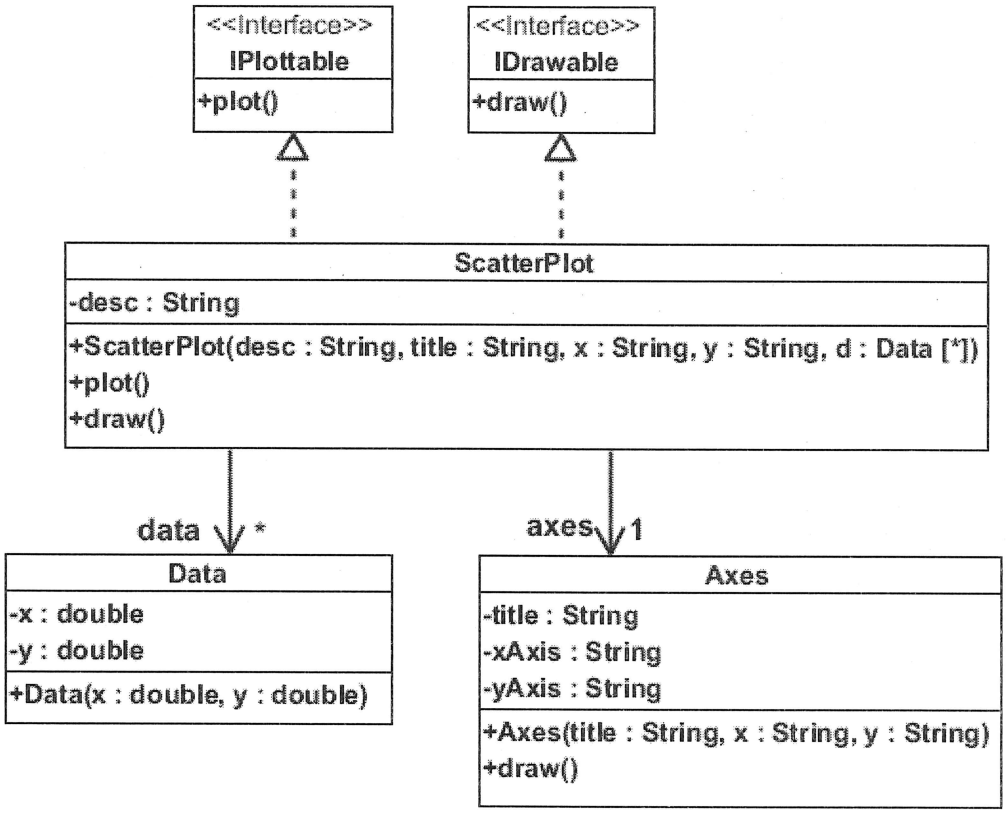
Use TWO of the SOLID design principles to explain how they were applied.

(10 marks)

APPENDIX A: Sequence Diagram



APPENDIX B: Class 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.