The University of Nottingham Ningbo China

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, AUTUMN SEMESTER 2021-2022

DEVELOPING MAINTAINABLE SOFTWARE

Time allowed: **Sixty (60) Minutes (One Hour)**

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced

Answer All SEVEN questions

Total Marks Available: 50

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn examination paper over until instructed to do so

INFORMATION FOR INVIGILATORS:
Collect both the exam papers and the answer booklets at the end of the exam.

SECTION A: Object-Oriented Concepts, Maintenance Principles and UI Design

Question 1: The basic object-oriented programming (OOP) is based on the concept of abstraction, encapsulation, inheritance, polymorphism, and interface.

a) Describe the concept of encapsulation and how it is useful in software maintenance. **[6 marks]**

b) What are the **THREE** ways to overload a method? Provide a code example to illustrate each way. **[9 marks]**

c) What is tagging interface? What are the **TWO** basic design purposes of a tagging interface? **[6 marks]**

Question 2: Gradle and Maven are popular build automation tools available for Java. Discuss **FOUR** key differences between Gradle and Maven. **[16 marks]**

Question 3: Animation is the basic class in JavaFX to define high-level animation. Animation is not restricted to movement and changing the background of a node over time is also considered an animation.

a) Describe **FOUR** key concepts involved in JavaFX timeline animation. **[8 marks]**

b) A client has requested you to code an animated square with the following conditions: **[11 marks]**

i. The square has the width of 120 pixels, starting at the position (25, 175).

ii. The final position of the square is at (145, 40) for a complete cycle of animation with ease in interpolator.

iii. The square should start with an opacity value of 1 and end with the value of 0.2.

iv. The total duration of the animation is 13 seconds.

v. The number of repeats for animation is 5 times.

Write a code snippet in JavaFX that satisfy the above requirements. Note that you do not need to define complete class(es) and method(s).

End of Section A: Total 56 marks

SECTION B: Design Principles and Refactoring

Question 4: Examine the code below and answer the following questions. Assume that a month has exactly 30 days, and more than 10 days of borrowing will incur a fine of 20% for `BookRental` and 50% for `MovieRental`.

```
class BookRental {
      String btitle; // book title
      String author;
      int day; // day of rental
      int month; // month of rental
      double rentalFee;

      double getTotalFee(int this_day, int this_month) {
            // Simplified logic to check overdue
            Assert this_month >= month;
            if((this_month – 1)* 30 + this_day > (month – 1) * 30 + day + 10)
                  return rentalFee * 1.2;
            else return rentalFee;
      }
}
class MovieRental {
      String mtitle; // movie title
      int classification;
      int day; // day of rental
      int month; // month of rental
      double rentalFee;

      double getTotalFee(int this_day, int this_month) {
            // Simplified logic to check overdue
            Assert this_month >= month;
            if((this_month – 1)* 30 + this_day > (month – 1) * 30 + day + 10)
                  return rentalFee * 1.5;
            else return rentalFee;
      }
}
```

Point out **FOUR** code smells in the code shown above. For each code smell, briefly describe the step(s) you would take to refactor. You do not have to write down the complete code, but it is sufficient to only describe the procedure. [**16 marks**]

Question 5: What are the essential steps to apply Singleton pattern to a class in Java? **[6 marks]**

Question 6: Describe Liskov's Substitution Principle and provide an example that violates this principle. Note: You don't have to provide the example code, but only need to describe the idea. **[5 marks]**

Question 7: A fleet management company hires you to design a "route planning" sub-system, which should fulfill the following design requirements:

- The company conducts business in China and Europe. The company has planning to expand its business to America, Oceania, etc.
- The route planner calculates the basic cost (HR cost, tax, etc.) for delivering the goods from one place to another. Based on the region, the route planner calculates the basic cost differently. For example, transporting in Europe contains both HR cost and tax while in China contains only HR cost.
- On top of basic cost, the route planner should calculate the transportation cost to be added to the basic cost. The transportation cost for one delivery could be the railway cost if delivered by train, the highway cost if delivered by lorry, or the shipping cost if delivered by ship. It can also be the combination of several. For example, to deliver the goods from Munich to Liverpool, one option is by train from Munich to Rotterdam, followed by ship from Rotterdam to Liverpool.
- The route planner maintains a list of transporting vehicles, including lorries, trains, and ships. In Europe, all vehicles are applicable, but in China, the company does not offer Sea transportation service.

Draw the class diagram of your design that fulfills the abovementioned requirements. In your design, apply the *two most suitable design patterns* from the four candidates: Factory Method, Strategy, Decorator, and State. **[17 marks]**

End of Section B: Total 44 marks