The University of Nottingham Ningbo China

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, AUTUMN SEMESTER 2023-2024

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 SIX 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 and Maintenance Principles

Question 1: Fundamental Object-Oriented Programming (OOP) is founded on the principles of abstraction, encapsulation, inheritance, polymorphism, and interface.

a) Explain the concept of encapsulation and its significance in software maintenance.

[6 marks]

b) Define the Java collections framework and outline two of its advantages.

[6 marks]

Question 2:

a) Define and elaborate the concept of Interface Segregation Principle in Object-Oriented Programming (OOP).

[6 marks]

b) Apply the Interface Segregation Principle to refactor the following Java code.

[8 marks]

```
public interface ILibraryRepository
{
    void AddBook(Book book);
    void EditBook(Book book);
    void DeleteBook(Book book);

    void AddCategory(Category category);
    void EdiCategory(Category category);
    void DeleteCategory(Category category);

    IList<Book> GetAllBooks();
    IList<Book> GetAllBooks(Book book);

    IList<Category> GetAllCategory();
    IList<Category> GetAllCategory(Category category);
}

public class BookRepository implements IBookRepository
{
    …
}
```

End of Section A: Total 26 marks

SECTION B: Design Principles and Refactoring

Question 3: Explain the term *Structural Design Pattern*, and provide the names and brief descriptions of two design patterns that belongs to this type. Your answer should not exceed 100 words in total.

[6 marks]

Question 4: Multiple choice questions – choose one or more correct answers.

a. Which of the actions below is/are not refactoring that removes the code smell?

A. Replace a parameter with a method named after it.

B. A comment becomes the name of a method.

C. Place the result of an expression in a local variable for later usage.

D. Replace polymorphism with switch, when distinguishing between different objects.

E. All of above.

b. Which of the following statements is/are true?

A. Refactoring may improve the functionality of the program itself.

B. Code refactoring is applied to remove malfunctioning of current code.

C. Refactoring should still be performed in small chunks of code.

D. It is trivial to create a set of unit tests for the existing code before refactoring.

E. None of above.

[6 marks]

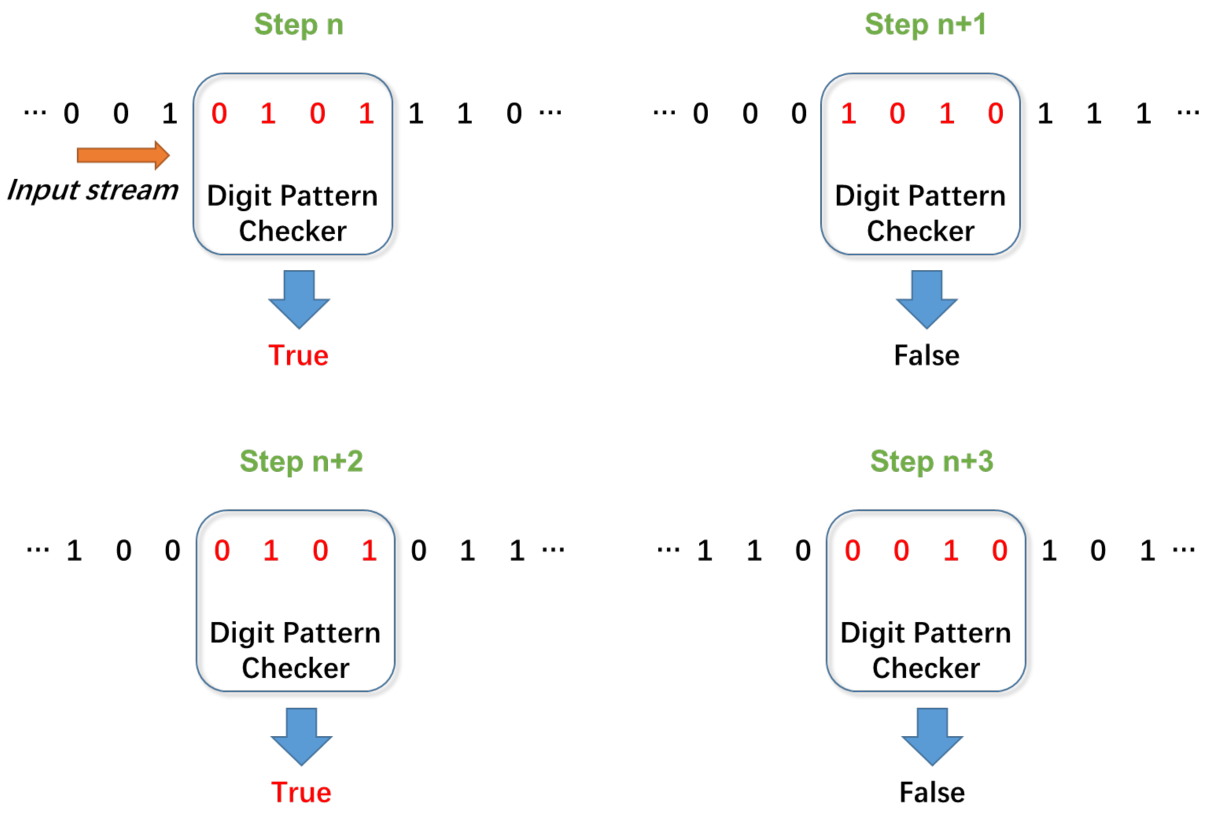
Question 5: A serial pattern checker is a system that keeps checking the serial digit pattern input as a stream into the system. The input stream is composed of serial random digits, namely 0s and 1s. The digit pattern is a fixed digit sequence, e.g., 0101. Once the system captures this pattern, it should output a positive signal, such as a Boolean 'true' to indicate that. At other times, it output a negative signal, such as a Boolean 'false' to indicate that the current sequence is not the pattern. The figure below shows an example of the system checking on pattern '0101'. Note that the digit stream flows from left to right. With the above description, answer the two questions below.

1. Design a state machine for the Digit Pattern Checker system to recognize the digit pattern '011010'. The state machine should contain exactly six (6) states. Draw the state machine, with properly designed states and transitions.

[9 marks]

2. Assume that you use Java to implement this system. Design your program with the State Pattern, by drawing the class diagram. Note: You are not required to write the code. Only the class diagram with State Pattern is expected.

[3 marks]



End of Section B: Total 24 marks