

UNIVERSITI TUNKU ABDUL RAHMAN

ACADEMIC YEAR 2017/2018

APRIL EXAMINATION

**UECS2344 SOFTWARE DESIGN**

WEDNESDAY, 2 MAY 2018

TIME : 2.00 PM – 4.00 PM (2 HOURS)

BACHELOR OF SCIENCE (HONS) SOFTWARE ENGINEERING

**Instructions to Candidates:**

This paper consists of two sections: Section A and Section B.

Answer all 3 questions in Section A.

Answer any 1 out of the 2 questions in Section B.

All questions carry equal marks.

UECS2344 SOFTWARE DESIGNSection A: Answer ALL questions in this Section.

- Q1. Consider the following program (Figure 1.1) written in C++ using the structured or procedural paradigm. The payroll.h file contains details of the structures, constants, and functions used in the program.

```
#include "payroll.h"
int main(void)
{
    EMPLOYEE_DATA emp_record;
    CHANGE_DATA changes;
    PAY_DATA pay_record;

    for (int i=1; i<=NUMBER_OF_EMPLOYEE; i++)
    {
        emp_record = read_emp_data();
        changes = read_update();

        if (changes.empty)
            cout << "No updates for employee" << endl;
        else
            update_emp_data(emp_record, changes);

        pay_record = compute_pay(emp_record)
        display_pay(pay_record)
    }

    return 0;
}
```

Figure 1.1

- (a) A Structure Chart can be used to show the functional decomposition of the program. Draw a Structure Chart for the program above. Include all Data Flows. (13 marks)
- (b) Functional decomposition is guided by the concept of cohesion. What is cohesion? State and elaborate the design guideline with respect to cohesion. (3 marks)
- (c) Another guiding concept is coupling. What is coupling? State and elaborate the design guideline with respect to coupling. (3 marks)
- (d) Suppose programmer A wrote the code for the main() function above and programmer B wrote the code for the remaining functions. This is possible because of procedural abstraction. What is procedural abstraction? How is procedural abstraction applied in this situation and what advantage does it provide? (6 marks)

[Total : 25 marks]

UECS2344 SOFTWARE DESIGN

Q2. Consider the partial Java code (Figure 2.1) given below.

```

public class BillingUI {
    private Controller controller;

    public void generateBill() {
        . . .
        Bill bill = controller.createBill(customerID);
        . . .
    }
    . . .
}

public class Controller {
    private CustomerList customerList;
    private Customer currentCustomer;
    . . .
    public Bill createBill(int customerID) {
        currentCustomer = customerList.find(customerID);
        Bill bill = currentCustomer.createBill();
        return bill;
    }
    . . .
}

public class CustomerList {
    . . .
    public Customer find(int customerID) {
        . . .
        return customer;
    }
    . . .
}

public class Customer {
    . . .
    private Bill bill;

    public void createBill() {
        bill = new Bill(fees, discountRate);
        return bill;
    }
    . . .
}

public class Bill {
    . . .
    public Bill(double fees, int discountRate) {
        . . .
        if (discountRate != 0)
            discount = computeDiscount();
    }
    public double computeDiscount() {
        . . .
    }
    . . .
}

```

Figure 2.1

**UECS2344 SOFTWARE DESIGN**Q2. (Continued)

- (a) Draw a Design Class Diagram which corresponds to the Java code given above, showing all the classes, and associations or dependencies. Also include the attributes and their data types.

*Note:*

1. You must include all relevant details given in the program.
2. You need NOT include the operations.

(7 marks)

- (b) Draw a Sequence Diagram which corresponds to the Java code given above, beginning with the call to the `generateBill()` method in the `BillingUI`.

*Note:*

1. You must include all relevant details given in the program.
2. You may ignore parts where details are not given.
3. You must show return messages, with labels where appropriate.

(18 marks)

[Total : 25 marks]

- Q3. (a) Consider the Analysis Class Diagram (Figure 3.1) below.

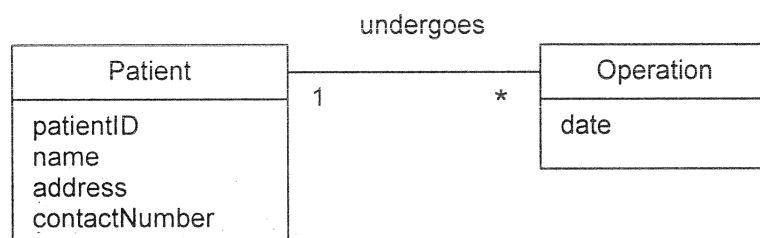


Figure 3.1

Draw a Design Class Diagram for the cases below.

*Note:*

1. You may assume any reasonable data types for the attributes.
2. You need NOT include the operations.

- (i) The navigability of the relationship is uni-directional from Patient to Operation. (4 marks)
  - (ii) The navigability of the relationship is bi-directional. (4 marks)
- (b) When developing a Design Class Diagram, three types or categories of object are usually determined. What are the three (3) types of object and what are their characteristics? (6 marks)

**UECS2344 SOFTWARE DESIGN****Q3. (Continued)**

- (c) Object persistence may be handled by using a relational database. Given the Analysis Class Diagram (Figure 3.2) below, show the relational database tables that the classes and relationships will be mapped to. Clearly indicate the primary and foreign keys in the tables.

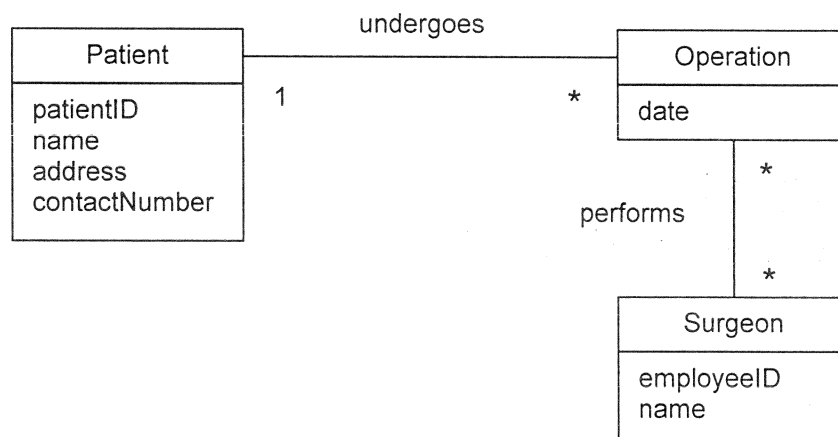


Figure 3.2

(11 marks)

[Total : 25 marks]

UECS2344 SOFTWARE DESIGNSection B: Answer ANY ONE question in this Section.

- Q4. (a) Analyse the Java code and corresponding output (Figure 4.1) below.
- (i) Identify the design pattern that has been applied. (5 marks)
  - (ii) What situation does this design pattern generally address? (4 marks)
  - (iii) Draw a class diagram (showing only class names and relationships) to represent the code and identify the role the classes play with respect to the design pattern. (6 marks)

```
public interface Element {  
    public abstract int getSize();  
    public abstract void addElement(Element element);  
}  
  
public class SingleElement implements Element {  
    private int size;  
    public SingleElement(int size) {  
        this.size = size;  
    }  
    public int getSize() {  
        return size;  
    }  
    public void addElement(Element element) {  
    }  
}  
  
public class MultipleElement implements Element {  
    private ArrayList<Element> elements;  
    public MultipleElement() {  
        elements = new ArrayList<Element>();  
    }  
    public void addElement(Element element) {  
        elements.add(element);  
    }  
}
```

Figure 4.1

UECS2344 SOFTWARE DESIGN

Q4 (a).(Continued)

```

    public int getSize() {
        int totalSize = 0;

        for (int i=0; i< elements.size(); i++) {
            Element element = (Element)elements.get(i);
            totalSize += element.getSize();
        }

        return totalSize;
    }
}

public class ElementTest {

    public static void main(String[] args) {

        Element one = new SingleElement(20);
        Element two = new SingleElement(30);

        Element multiple1 = new MultipleElement();

        multiple1.addElement(one);
        multiple1.addElement(two);

        System.out.println("Size of multiple1 is "
            + multiple1.getSize());

        Element three = new SingleElement(5);
        Element four = new SingleElement(10);

        Element multiple2 = new MultipleElement();

        multiple2.addElement(three);
        multiple2.addElement(four);
        multiple2.addElement(multiple1);

        System.out.println("Size of multiple2 is "
            + multiple2.getSize());
    }
}

```

**Output**

Size of multiple1 is 50  
 Size of multiple2 is 65

Figure 4.1 (Continued)

**UECS2344 SOFTWARE DESIGN****Q4. (Continued)**

- (b) Design patterns are a form of reuse. Frameworks are another form of reuse. What are frameworks? How do frameworks support reuse? (10 marks)

[Total : 25 marks]

- Q5. (a) What architectural pattern does the following diagram (Figure 5.1) illustrate? Identify the characteristics of the architectural pattern and justify your answer.

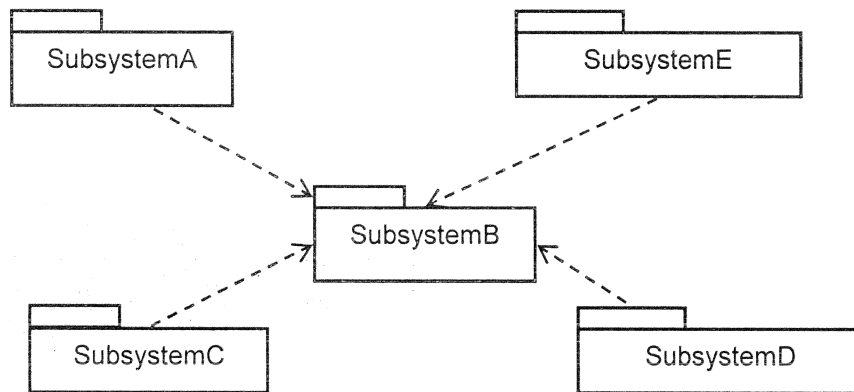


Figure 5.1

(10 marks)

- (b) Consider the following diagram (Figure 5.2) which shows some classes organized in different packages. Is this an application of layering or partitioning? Differentiate between the two and justify your selection.

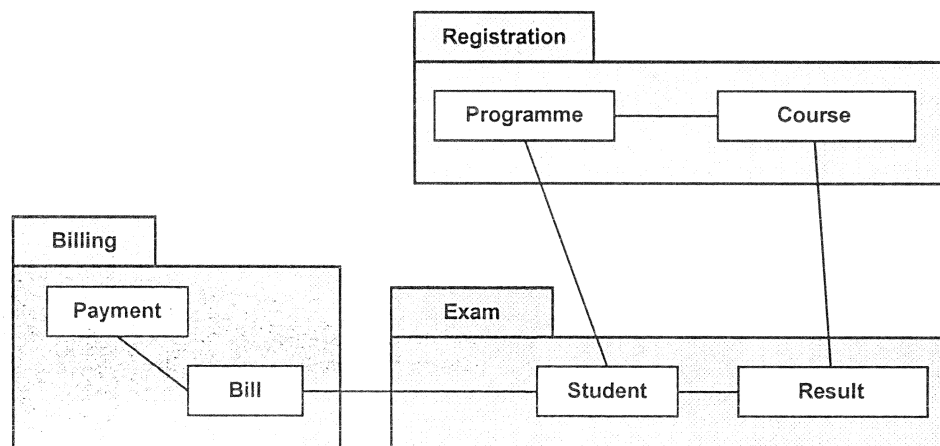


Figure 5.2

(10 marks)

- (c) A recent trend in software development is the Model-Driven Architecture (MDA). What are the main characteristics of the MDA? (5 marks)

[Total : 25 marks]