



UNIVERSITY OF MORATUWA
Faculty of Information Technology

Bachelor of Information Technology (BIT)
Level 2 – Semester 1 Examination

ITE 2132 - OBJECT ORIENTED PROGRAMMING

Time Allowed: 3 hours

September 2020

INSTRUCTIONS TO CANDIDATES

1. This paper contains 5 questions on 5 Pages.
2. The total marks obtainable for this examination is 100. The marks assigned for each question & sections thereof are included in square brackets.
3. This examination accounts for 60% of the module assessment.
4. This is a closed book examination.
5. Calculators are not allowed.
6. Answer **All** questions.

ADDITIONAL MATERIAL

None

Continued...

Question 1 [20 Marks]

- (a) Briefly explain the terms Encapsulation, Inheritance and Polymorphism in relation to Object-Oriented programming. [4 Marks]
- (b) Write down two differences between a Constructor and a Method. [4 Marks]
- (c) Write Java methods to perform the following.
- i) To receive two integers and return their sum of squares. Suggested name for the method: *"sumOfSquare"*.
 - ii) To receive three integers and return the largest value. Suggested name for the method: *"largest"*.
 - iii) To receive the base and height of a triangle in double type values and return its area. Suggested name for the method: *"triangleArea"*.
 - iv) To receive an array of integer and return the sum of all the integers in it. Suggested name for the method: *"sumArray"*.

[3 x 4 = 12 Marks]

Question 2 [20 Marks]

- (a) Explain the terms Java Development Kit (JDK) and Java Runtime Environment (JRE). [4 Marks]
- (b) Briefly explain the role of Java Virtual Machine (JVM). [4 Marks]
- (c) What is the output of the following Java program? Clearly state the reasons, if it provides an error message.

```

class Parent {
    private int x;
    private int y;

    double addition() {
        return (x + y); }
}

class Child extends Parent {
    String name;
    String address;

    Child(String n, String address){
        name = n;
        address = address; }

    Child(String n){
        this(n); }
}

```

[6 Marks]
Continued...

- (d) What is the output of the following Java program? Clearly state the reasons, if it provides an error message.

```
class Student {

    final void calcAverage(double mark1, double mark2) {
        double avg =(mark1 +mark2)/2.0;
        System.out.println("Avg Mark of Student:" + avg);
    }
}

class PartTimeStudent extends Student {

    String name;

    @Override
    final void calcAverage(double mark1, double mark2) {
        double avg =(mark1 +mark2)/2.0;
        System.out.println("Avg Mark of Part Time Student:" + avg);
    }
}

class Question {

    public static void main(String[] args) {
        PartTimeStudent pt =new PartTimeStudent();
        pt.calcAverage(50, 30);
    }
}
```

[6 Marks]

Question 3 [20 Marks]

You have been asked to develop a software system to “Healthy Heart” Hospital. The system manages information about Doctors. Doctors can be of two categories, namely General Practitioners and Specialists. Three classes called **Doctor**, **GeneralPractitioner** and **Specialist** exist in the application.

A Doctor has name, address, phone number, and registration number as properties. A General Practitioner has name, address, phone number, registration number and hourly rate as properties. A Specialist has name, address, phone number, registration number and doctor's charge per patient as properties.

At the end of every month, the system must calculate the total payment for each doctor. Thus, you have decided to add a method called **totalPayments()** to the doctor class. Method **totalPayment()** should take an integer parameter and return the total payment for the doctor. General Practitioners are paid based on the number of visiting hours during the month and Specialists are paid based on the number of patients consulted during a month. Total payment for a General Practitioner is equal to the multiplication between hourly rate and the number of visiting hours during the month. You may take the number of visiting hours as a parameter. Total payment for a Specialist is equal to the multiplication between doctor charge and the number of patients consulted during the month. You may take the number of patients consulted as a parameter.

Continued...

Answer the questions (a) to (d) based on the above scenario.

- (a) Is **Doctor** class, a final class or an abstract class? Justify the answer.
[4 Marks]
- (b) Write the **Doctor** class implementation. Include at least 2 constructors to the class. Use the keyword *this* in appropriate places.
[4 Marks]
- (c) Write the **GeneralPractitioner** class implementation. You should include at least one constructor to the class which will set the values of parameters to the class members and must override the **totalPayment()** method. Use the keywords *this* and *super* in appropriate places.
[6 Marks]
- (d) Write the **Specialist** class implementation. You should include at least one constructor to the class which will set the values of parameters to the class members and must override **totalPayment()** method. Use the keywords *this* and *super* in appropriate places.
[6 Marks]

Question 4 [20 Marks]

- (a) Briefly explain the advantages of a UML Use Case diagram in software modeling.
[4 Marks]
- (b) Explain the terms *include* and *extend* in Use Case diagrams using suitable examples.
[4 Marks]
- (c) Consider the following scenario and draw a UML Use Case diagram to represent the functionality of the ABC Food system. Your diagram should capture all the relevant actors, use cases, and required relationships.

ABC Food is a small business that sells sweets and cakes. The owner of ABC Food wants to have a software system to make the current business more efficient and easier to handle. The owner buys sweets and some cakes from suppliers to increase the range of products for sale. The shop operates with a sales assistant, two bakers, and the owner himself. Sometimes owner himself play the role of sales assistant if the sales assistant is not available.

When customers buy items, the sales assistant makes the transaction. When the sales assistant enters the items and quantities, system should display the invoice. Payments can be made through cash or cards and a printed receipt is provided to the customer for each sales transaction. At the end of each sales transaction, stocks should be updated to reflect the sale.

The owner reviews the sales of the items by studying the daily sales report. The bakers can view the baking requests through a tablet. Baker has to decide and enter the amounts of raw ingredients required for that request.

[12 Marks]

Continued...

Question 5 [20 Marks]

- (a) Explain the main components of UML Activity Diagram using examples. [4 Marks]
- (b) Draw a UML Activity diagram with relevant swim lanes for the “*Pay and exit from car park*” scenario given below.

Main success activities:

1. Use case starts when user clicks on the 'Pay & Exit ' button in the payment machine
2. The system requests the vehicle number
3. The user enters the vehicle number
4. System locates the car
5. System shows the image of the car to the user
6. User identifies the car and confirms
7. System shows the ticket information including parking charges
8. User enters card details to make payment
9. System sends the card details to payment gateway to validate
10. System prints the receipt
11. User collects the receipt and drives to the Exit gate
12. System identifies the car at the Exit gate
13. System opens the Exit gate for the user
14. Use case ends when the user exits the car park

Extensions:

- 4a. if the system cannot locate the car, go back to the step 2
- 6a. if the user does not identify the car image, go to step 4
- 9a. if the card details invalid, display a relevant message and go to step 8

[16 Marks]

END OF PAPER

