TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

ACADEMIC YEAR 2024/2025

MAY/JUNE EXAMINATION

BACS2023 OBJECT-ORIENTED PROGRAMMING

MONDAY, 26 MAY 2025 TIME: 9.00 AM – 11.00 AM (2 HOURS)

BACHELOR OF COMPUTER SCIENCE (HONOURS) IN INTERACTIVE SOFTWARE TECHNOLOGY

BACHELOR OF INFORMATION SYSTEMS (HONOURS) IN ENTERPRISE INFORMATION SYSTEMS

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN INFORMATION SECURITY
BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN INTERNET TECHNOLOGY
BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN SOFTWARE SYSTEMS DEVELOPMENT

BACHELOR OF SCIENCE (HONOURS) IN MANAGEMENT MATHEMATICS WITH COMPUTING

BACHELOR IN DATA SCIENCE (HONOURS)

BACHELOR OF SOFTWARE ENGINEERING (HONOURS)

Instructions to Candidates:

Answer **ALL** questions. All questions carry equal marks.

Question 1

a) Discuss **TWO (2)** advantages of Object-Oriented Programming. (9 marks)

b) Explain the usage of the following access modifiers/keywords and provide an example of data for each of them:

(i) static (4 marks)

(ii) final (4 marks)

c) Distinguish **TWO (2)** differences between method *overloading* and method *overriding* in Object-Oriented Programming with suitable Java codes. (8 marks)

[Total: 25 marks]

Question 2

a) Access modifiers are used to specify the visibility or accessibility of classes, methods, variables, and constructors. Define and analyse the scope of the **FOUR (4)** main access modifiers in Java. (12 marks)

b) (i) Explain the meaning of ***Class*** in the context of Object-Oriented Programming. (2 marks)

(ii) Refer to the following Unified Modelling Language (UML) class diagram in Figure 1:

Ball
- colour : String - inch : double
+ Ball (colour: String, inch: double) + setColour (colour: String): void + setInch (inch: double): void + getColour () : String + getInch () : double

Figure 1

Develop a `Ball` class that models a ball object based on the above diagram. You are required to follow the Object-Oriented Programming principles. (8 marks)

(iii) Create an instance of a `Ball` object named `football`. Set the football colour to brown and size to 27-inch. (3 marks)

[Total: 25 marks]

Question 3

In a university, employees can be classified into two broad categories:

- **Professor** - Professors are employees who handle academic and research activities. They have specific attributes like their field of expertise and the department they belong to.
- **Administrative Staff** - Administrative staff refers to non-academic employees such as office assistants, security personnel, and finance staff. They perform duties that support the operational and management functions of the university.

Both Professors and Administrative Staff share some common attributes such as **name, employee ID, contact number**, and **email**. However, they also have their specific attributes, for examples:

- **Professors**: field of expertise, department.
- **Administrative Staff**: job title (e.g., office assistant, finance officer), office location.

a) Apply Object-Oriented programming principles in designing the following classes. Ensure these classes support the mechanism of **inheritance** and **encapsulation**. (Ignore constructors, setters and getters)

(i) Employee (3 marks)

(ii) Professor (3 marks)

(iii) Administrative Staff (3 marks)

b) Construct a simplified UML class diagram for Employee, Professor and Administrative Staff constructed in Question 3 a). (5 marks)

c) Create a polymorphic array consisting of **ONE (1)** professor and **ONE (1)** administrative staff. (5 marks)

d) Assume the Employee, Professor and Administrative staff class have been created, and their relationship has been formed. Analyse the validity of the following statements with justification:

(i) Professor pro = new AdministrativeStaff(); (3 marks)

(ii) Employee emp = new Professor (); (3 marks)

[Total: 25 marks]

Question 4

A Smart Home Automation System is designed to control different electronic devices, such as Lights, Fans, and Air Conditioners. These devices shared some common methods: `turnOn()`, `turnOff()`, etc, however, they are implemented differently.

a) (i) Provide a reason of using of *Interface* in JAVA based on above scenario. (2 marks)

(ii) Develop the *Interface* named `SmartDevice` that complies with above scenario. (3 marks)

b) (i) Develop the `Light` class. The following requirements must be fulfilled:

- A `boolean` variable `isOn` that keeps track of whether the light is on or off.
- A constructor that accepts a `boolean` parameter that set the initial state of the light.
- Display the message **"Light is now ON."** when the light is turned on.
- Display the message **"Light is now OFF."** when the light is turned off.
- Ensure that the `turnOn()` and `turnOff()` methods correctly update the `isOn` variable when the light is switched on or off.

(12 marks)

(ii) Create an instance of the `Light` class with its initial state set to **Off**. Then, turn it on using the `turnOn()` method. (4 marks)

c) Discuss **TWO (2)** differences between an `abstract` class and an `interface`. (4 marks)

[Total: 25 marks]