Faculty of Computing
Year 1 Semester 2 (2025)

SE1020 – Object Oriented Programming Lab Sheet 09

Question 1: Transport Fare System (Abstract Classes)
Design a system that calculates transport fares based on the type of transport.

(a) Create an abstract class **Transport** with attributes vehicleNumber and distanceTravelled.

(b) Include an abstract method called ***calculateFare()*** and a concrete method ***displayDetails()*** to show vehicle information.

(c) Create two subclasses as follows:

  A. Class **Bus** to calculate the fare as:
  Bus fare = 10 × distance

  B. Class **Taxi** to calculate the fare as:
  Taxi fare = 20 × distance + 50 (base fee)

(d) Override ***calculateFare()*** in each subclass and update ***displayDetails()*** to include fare details.

(e) In the `main` method, create objects for both 'Bus' and 'Taxi', and display details.

Question 2: Printing Management System (Interfaces)
Build a print system where different items can be printed using a common interface.

(a) Create an interface **Printable** with a method ***printDetails().***

(b) Create two classes called **Report** and **Invoice**, both implementing Printable interface in each class.

(c) Each class should have suitable attributes (e.g., title for Report, invoiceId for Invoice) and constructor.

(d) Implement ***printDetails()*** to show the specific content in each class.

(e) In the main method, create two objects per Report and Invoice classes, store them in a Printable[ ] array and use a loop to print.

Question 3: University System (Static Modifiers)

Develop a system to manage student information and university details.

(a) Create a **Student** class with attributes studentId, studentName.

(b) Include a static variable universityName to be shared among all students.

(c) Add constructors and a method ***displayStudentInfo()*** that includes university name.

(d) In the main method:

A. Set universityName using the class name.

B. Create a three Student objects and display their details.

C. Change the university name mid-way and show its impact on all objects by calling the ***displayStudentInfo()*** again .

Question 4: Device Inventory System (Static Block & Static Method)

Track the number of devices registered in an inventory.

(a) Create a class **Device** with attributes deviceId, deviceType.

(b) Declare a static variable deviceCount.

(c) Use a **static block** to initialize deviceCount to 100.

(d) Each time a Device object is created, increment the counter.

(e) Include a static method ***displayDeviceCount()*** to show the current count.

(f) In the main method, create 3 devices and display their info and device count.

Question 5: Employee Payroll System (Abstract Class + Interface)

Design a payroll system where employees calculate their salary differently but follow a

common interface.

(a) Create an abstract class **Employee** with empId, name, and abstract method ***calculateSalary().***

(b) Create an interface **Payable** with method ***generatePayslip().***

(c) Two concrete classes:

A. **PermanentEmployee** → fixed monthly salary.

B. **ContractEmployee** → paid hourly.
Both classes should implement Payable and extend Employee.

(d) In the main method, create both types of employees and print their salary and payslip.

Question 6: E-Learning Platform (Interface + Multiple Implementation)
Model an e-learning system where users can enroll in courses or teach.

(a) Create two interfaces:
   A. Called **Enrollable** with enrollCourse(String courseName) as an attribute
   B. Called **Teachable** with assignCourse(String courseName) as an attribute

(b) Create a class called **Student** which implements Enrollable interface

(c) Create a class called **Instructor** which implements both Enrollable and Teachable interfaces

(d) Create a main method that demonstrates a student enrolling in a course and an instructor both enrolling and assigning courses.