UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Department of Computer Science
Faculty of Computing

SECPH - BACHELOR OF COMPUTER SCIENCE (DATA ENGINEERING)

MINI PROJECT OOP REPORT

TITLE: STUDENT COURSE REGISTRATION SYSTEM

PREPARED BY: Privic (GROUP)

NAME	MATRIC NO
MUHAMMAD SYAHMI FARIS BIN RUSLI	A23CS0138
AHMAD ADIB ZIKRI BIN A.MAZLAM	A23CS0205

DATE:

28th JUNE 2025

PREPARED FOR:

DR. KHATIBSYARBINI

1.0 INTRODUCTION

The purpose of this mini project is to apply and demonstrate the core principles of Object-Oriented Programming (OOP) learned throughout the course, specifically those covered in Chapters 5 to 9. As students, we are often required to register for courses each semester, and this repetitive process presents a perfect use case to simulate through programming. For this reason, we decided to create a **Student Course Registration System** that closely mirrors the basic functionality of an academic course registration portal.

The project was implemented using Java and focuses on providing a structured, menu-based, and interactive command-line application. The system allows a student to register either as an **Undergraduate** or **Postgraduate** student, input personal details, and perform course management functions such as registering and dropping subjects.

Through this mini-project, we aimed not only to build a functional system but also to strengthen our understanding of how OOP concepts work together in real-world application design. The system is modular, scalable, and easy to maintain, just like professional software should be.

2.0 BRIEF PROJECT DESCRIPTION

The Student Course Registration System is a console-based Java program that allows:

- Students to register by providing their name, matric ID, and either year of study (for undergraduates) or thesis title (for postgraduates).
- Course registration by entering a course code and title.
- Viewing of student information and registered courses.
- Dropping of courses already registered\

Only **Undergraduate students from Year 1 to Year 3** are allowed to register, with validation that prevents invalid entries. The system demonstrates clean class design using Java OOP features and provides a menu-driven interface to interact with different functionalities.

The system also integrates abstraction, inheritance, polymorphism, aggregation, association, and exception handling across different modules.

3.0 CHAPTER WISE IMPLEMENTATION

Chapter 5: ArrayList

- The **ArrayList<Course>** is used in the Student class to store a dynamic list of registered courses.
- Courses are added via .add() and removed via .remove(), supporting unlimited course registration until manually dropped.

Chapter 6: Class Relationships

Association:

- Student class holds a list of Course objects showing that a student can **register multiple courses**.

Aggregation:

- Student has a Name object composed of firstName and lastName.
- This demonstrates a **has-a** relationship where the Name object can exist independently.

Chapter 7: Inheritance

- User is an abstract superclass that contains the student ID and abstract method printInfo().
- Student extends User and is also abstract.
- Undergraduate and Postgraduate are concrete subclasses that implement their own printInfo() method, showing specialisation.

Chapter 8: Polymorphism

- printInfo() is **overridden** in both Undergraduate and Postgraduate.
- The Student reference is used to store either subclass at runtime, enabling **dynamic binding**.

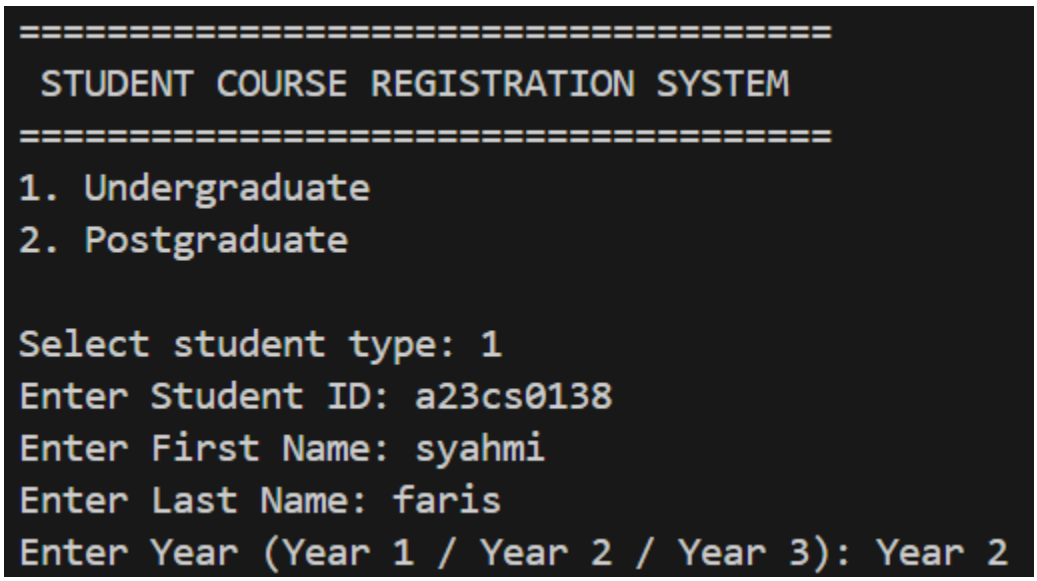
Chapter 9: Exception Handling

- The program uses try-catch blocks to manage invalid inputs such as:
  - Invalid student type (not 1 or 2)
  - Invalid year (must be Year 1–3 for undergraduates)
  - Empty course code/title during registration
- When invalid input is detected, error messages are displayed and the user is prompted again until valid input is provided.

4.0 SAMPLE OUTPUT / SCREENSHOT

4.1 STUDENT REGISTRATION MODULE

Valid Undergraduate Input



Valid Postgraduate Input

```
====================================
 STUDENT COURSE REGISTRATION SYSTEM
====================================
1. Undergraduate
2. Postgraduate

Select student type: 2
Enter Student ID: A23CS0205
Enter First Name: Adib
Enter Last Name: Zikri
Enter Thesis Title: Machine Learning
```

Invalid Student Type

```
Select student type: 3
Invalid student type. Please enter 1 or 2.
```

Invalid Year (Undergraduate)

```
Enter Year (Year 1 / Year 2 / Year 3): Year 4
Only Year 1 to Year 3 students can register courses. Please try again.
```

4.2 COURSE REGISTRATION MODULE

Valid Course Entry

```
====================================
              MAIN MENU
====================================
1. Register a Course
2. List Registered Courses
3. Drop a Course
4. View Student Info
5. Exit
Choose option: 1
Enter Course Code: SECV3233
Enter Course Title: Web Programming
Registered course: SECV3233 - Web Programming
```

Empty Course Code or Title

```
Choose option: 1
Enter Course Code:
Enter Course Title:

ERROR: Course code and title cannot be empty.
```

4.3 VIEW REGISTERED COURSES

Courses Exist

```
Choose option: 2

Courses Registered:
- SECV3233 - Web Programming
```

No Courses Registered

```
Choose option: 2
No courses registered.
```

4.4 DROP COURSE MODULE

Drop Existing Course

```
Choose option: 3

Registered Courses:

Courses Registered:
- SECP3233 - Data Engineering
Enter Course Code to Drop: SECP3233
Dropped course: SECP3233 - Data Engineering
```

Enter Invalid Course Code

```
Choose option: 3

Registered Courses:

Courses Registered:
- SECP3233 - Data Engineering
Enter Course Code to Drop: SECP3333
No course found with code: SECP3333
```

No Courses to Drop

```
Choose option: 3

Registered Courses:
No courses registered.
```

4.5 VIEW STUDENT INFO MODULE

Example (Undergraduate)

```
Choose option: 4

[Undergraduate Student]
ID    : A23CS0138
Name  : SYAHMI FARIS
Year  : Year 2
No courses registered.
```

4.6 EXIT MESSAGE

```
Choose option: 5

Thank you for using the system. Goodbye!
```

5.0 UML CLASS DIAGRAMS