DEPARTMENT Of PHYSICAL SCIENCE

FACULTY Of APPLIED SCIENCE

Name: **Introduction**	Assignment Number: **1**
Course: **IT 3113 (P)**	Date: September 18, 2025

Problem 1

People

- Alice is a student.
- Bob is a student.
- Charlie is a student.
- Diana is a student.
- Dr. Smith is a teacher.
- Dr. Jones is a teacher.
- Dr. Clark is a teacher.
- Dr. Williams is a teacher.
- Dr. Brown is a teacher.

Courses

- CS101 is a Programming course.
- CS102 is a Databases course.
- CS103 is an AI course.
- CS104 is a Maths course.
- CS105 is an Algorithms course.

Enrolments

- Alice is enrolled in CS101.
- Bob is enrolled in CS101.
- Charlie is enrolled in CS102.

- Diana is enrolled in CS103.
- Alice is also enrolled in CS104.
- Bob is enrolled in CS105.
- Charlie is also enrolled in CS104.

Teaching

- Dr. Smith teaches CS101.
- Dr. Jones teaches CS102.
- Dr. Clark teaches CS103.
- Dr. Williams teaches CS104.
- Dr. Brown teaches CS105.

Prolog Facts

```prolog
% --- People ---
student(alice).
student(bob).
student(charlie).
student(diana).

teacher(dr_smith).
teacher(dr_jones).
teacher(dr_clark).
teacher(dr_williams).
teacher(dr_brown).

% --- Courses ---
course(cs101, programming).
course(cs102, databases).
course(cs103, ai).
course(cs104, maths).
course(cs105, algorithms).

% --- Enrolments ---
enrolled(alice, cs101).
enrolled(bob, cs101).
enrolled(charlie, cs102).
enrolled(diana, cs103).
enrolled(alice, cs104).
enrolled(bob, cs105).
enrolled(charlie, cs104).

% --- Teaching ---
teaches(dr_smith, cs101).
teaches(dr_jones, cs102).
teaches(dr_clark, cs103).
teaches(dr_williams, cs104).
teaches(dr_brown, cs105).
```

Prolog Rule Exercises

Question 0.1. *Create a rule `classmate(X,Y)` which is true when X and Y are enrolled in the same course and X \= Y.*
Query: `?- classmate(alice,Y).`

Question 0.2. *Create a rule `is_student_of(Student,Teacher)` that succeeds when Teacher is an instructor of Student.*
Query: `?- is_student_of(charlie, dr_jones).`

Question 0.3. *Create a rule `share_teacher(X,Y)` that is true if two students have at least one teacher in common.*
Query: `?- share_teacher(alice,Y).`

Question 0.4. *Define a rule `beginner_course(C)` which is true if C is a course in programming or maths (use the ; OR operator).*
Query: `?- beginner_course(X).`

Question 0.5. *Create a rule `enrolled_in_any_course(Student)` that is true if a student is enrolled in any course (use the anonymous variable _).*
Query: `?- enrolled_in_any_course(bob).`

Question 0.6. *Create a rule `has_students(Teacher)` that is true if a teacher teaches at least one student (again use _).*
Query: `?- has_students(dr_smith).`

Question 0.7. *Define a rule `advanced_student(Student)` which is true if the student is enrolled in an advanced course (you already have `advanced_course/1`).*
Query: `?- advanced_student(X).`

Question 0.8. *Create a rule `teaches_multiple(Teacher)` that is true if a teacher teaches two different courses.*
Query: `?- teaches_multiple(dr_smith).`

Question 0.9. *Create a rule `not_enrolled(Student,Course)` which is true if the student is not enrolled in a course (use \+ operator for negation).*
Query: `?- not_enrolled(alice, cs103).`

Question 0.10. *Create a rule `student_pair(Student1,Student2,Course)` which gives pairs of students enrolled in the same course.*
Query: `?- student_pair(X,Y,cs101).`

Rules to Implement

```
% X and Y are classmates if they are enrolled in the same course
classmate(X,Y) :- enrolled(X,C), enrolled(Y,C).

% A student is student_of a teacher if the teacher teaches a course
% in which the student is enrolled
is_student_of(Student,Teacher) :-
    enrolled(Student,C),
    teaches(Teacher,C).

% Beginner course is either programming or maths
beginner_course(C) :-
    course(C,programming);
    course(C,maths).
```

Sample Queries

```
?- classmate(alice,Y).
?- is_student_of(alice,T).
?- beginner_course(C).
?- enrolled(_, cs104).  % who is enrolled in CS104?
?- teaches(T,_).        % list all teachers
```