

FAMILY NAME	
OTHER NAMES	
STUDENT ID	
SIGNATURE	

THE UNIVERSITY OF NEW SOUTH WALES

Semester, Year

COMP2511 Sample Exam

OBJECT-ORIENTED DESIGN AND PROGRAMMING

- (1) READING TIME – 10 MINUTES
- (2) TIME ALLOWED – 2 HOURS
- (3) TOTAL NUMBER OF PAGES – 2

- (4) THIS EXAMINATION COUNTS FOR 60% OF THE FINAL MARK
- (5) TOTAL NUMBER OF QUESTIONS – 4
- (6) ANSWER **ALL** QUESTIONS
- (7) ALL QUESTIONS ARE OF EQUAL WEIGHT

- (8) NO MATERIALS MAY BE TAKEN INTO THE EXAMINATION ROOM
- (9) THIS PAPER MAY **NOT** BE RETAINED BY THE CANDIDATE
- (10) INDICATE WHICH QUESTIONS HAVE BEEN ANSWERED IN EACH BOOKLET

ANSWERS MUST BE WRITTEN CLEARLY IN INK. EXCEPT WHERE THEY ARE EXPRESSLY REQUIRED, PENCILS MAY ONLY BE USED FOR DRAWING, SKETCHING OR GRAPHICAL WORK.

Question 1 Object-Oriented Design

Consider a university enrolments system with the following requirements.

- Students enrol in courses that are offered in particular semesters.
- Students receive grades (pass, fail, etc.) for courses in particular semesters.
- Courses may have prerequisites (other courses) and must have credit point values.
- For a student to enrol in a course, s/he must have passed all prerequisite courses.
- Course offerings are broken down into multiple sessions (lectures, tutorials and labs).
- Sessions in a course offering for a particular semester have an allocated room and timeslot.
- If a student enrolls in a course, s/he must also enrol in some sessions of that course.

- (a) Draw a UML class diagram for an object-oriented system to implement the above requirements and provide CRC cards for your main classes. *Clearly explain how your design distinguishes between a course and an instance of a course at a particular semester.*
- (b) Define a use case for a student enrolling in a course that has a prerequisite that s/he has passed, and then define a walkthrough demonstrating how this would be handled in your system.

Question 2 Programming by Contract

- (a) Design a **BankAccount** class for maintaining bank balances. Each bank account should have a current balance and methods implementing deposits and withdrawals; however money can only be withdrawn from an account if there are sufficient funds. Each account has a cash withdrawal limit of \$800 per day.

Give Java implementations of the methods in your design. *Explain how your implementation enforces the daily limit on withdrawals.*

- (b) Give preconditions and postconditions for each method in the **BankAccount** class and provide a rigorous argument that $balance \geq 0$ is a class invariant. *Explain how your code is consistent with your pre- and postconditions.*

Question 3 Generic Types and Polymorphism

- (a) Define a **Set<E>** interface type that can handle elements of a generic type **E**. Consider operations on set membership (adding and removing elements, testing membership), and basic operations on sets (union, intersection, subset). Include an equals method for sets.
- (b) Define a class realizing the **Set<E>** interface that uses an **ArrayList<E>** to store elements. *Explain how your code enforces the class invariant that all elements of the set are distinct.*

Question 4 Design Patterns

- (a) In a manufacturing plant, a product (such as a car or a computer) is assembled from other parts. For example, a car might be assembled from a chassis, an engine and a body; in turn, the chassis might be assembled from a frame and some wheels, etc.

Use the Composite pattern to write Java classes for **Assembly** and **Item** with a method `getPrice` for calculating the total price of any given part. Each **Item** has a given price (assume that `price` is a parameter to the class constructor for **Item**), and the price of an **Assembly** is the *total* price of all parts in the assembly.

Draw a UML class diagram for your program that demonstrates that your design conforms to the Composite pattern as defined in this course.