National University of Computer and Emerging Sciences

Formal Methods (SE2003)

Sessional-II Exam

Date: 9th April, 2025

Course Instructor(s)

Dr.Wafa Basit

Total Time: 1 Hours
Total Marks: 50
Total Questions: 3

Semester: SP-2025
Campus: Lahore
Dept: Software Engineering

_____ _____ _____ _____

Student Name Roll No Section Student Signature

_____ _____

Vetted by Vetter Signature

Instructions

- Make assumptions where necessary
- In case of multiple solutions, mention the final one
- All Questions have to be attempted on question paper.
- Use of lead pencil is not allowed.
- Cutting on the paper would result in deduction of marks.

Question #1: Fill in the table (10 Points)

Function name	Used for	Return type
Including (elem)		
oclIsTypeOf(t: Type)		
allInstances()		
count(elem)		
One(elem)		

Question #2: Fill in the blanks (10 Points)

1. The `@pre' postfix is allowed only in ----------.
2. A ----------expression is only known within its specific expression.
3. ----------is a type of Collection which is Non-ordered and unique.
4. “Whenever an instance of a class is expected,one can always substitute an instance of any of its subclasses.” It Is ----------Principle.
5. ---------- is the top type in OCL.

6. DLL stands for----------.
7. The ---------- invariants are added (logically "and"ed) to the class's own invariants.
8. Sum>10 is a valid precondition but Sum>0 is a----------precondition.
9. For the following class the invariant would be----------
   Class Line { //Line defined by two points p1(x1, y1), p2(x2, y2)
   Point p1, p2;}
10. DbC uses ---------- as a contract between supplier and client.

Question # 3 Fig. 1 presents a small, contrived example of a class model in UML for a simple system that supports scheduling of offerings of seminars to a collection of attendees by presenters who must be qualified for the seminars they present.

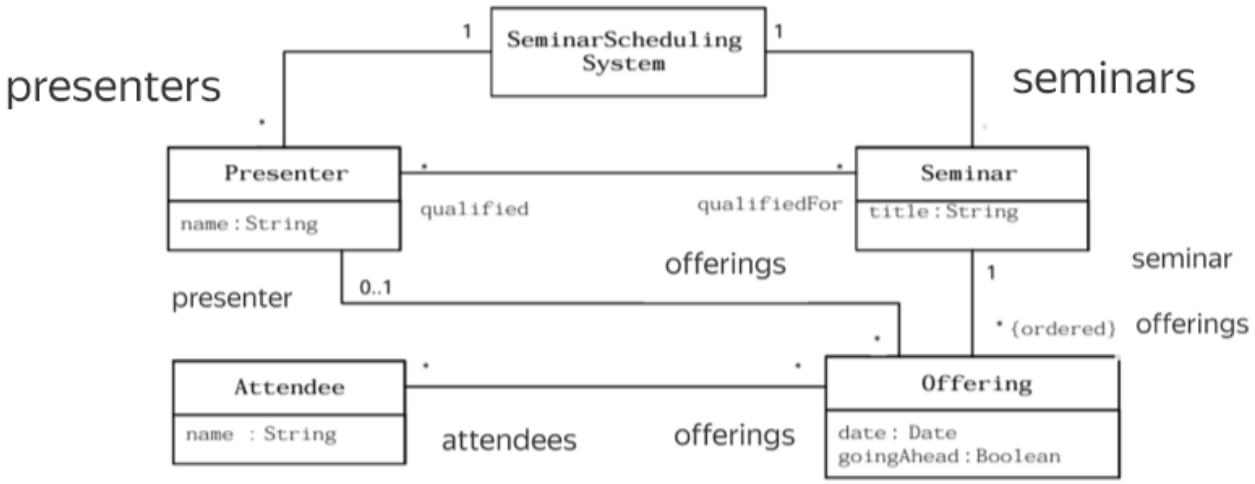

Fig. 1. A class diagram for a seminar scheduling system

Solve the following questions using the class diagram given above. (30 Points)

1. Invariant: The offerings of a Presenter is a subset of offerings of a seminar, he is qualified for. (5 Points)

2. Define a function named “ListAttendees” in the Offering class which returns the collection of attendees as a unique set. (5 Points)

3. Invariant: In the Seminar Scheduling System names of all Seminars and their presenters have unique names. (5 Points)

4. Create a derived attribute namely ‘OfferingName’ in the Offering class, which concatenates the name of the Presenter, hyphen and Seminar name. (Assume that Presenter is not null). (5 Points)

5. In the Seminar Scheduling system, one result of executing an operation *schedule* is the creation of a new offering. In order to assert that a new offering '*o'* is created, we need to assert that it did not exist prior to executing the operation but does exist after executing the operation. After the operation is executed the *o.Seminar* is assigned 's' and *o.date* is assigned *d.* The schedule operation doesn't add attendees and presenters, so they should be empty. The attribute goingAhead should also be true for the object *O.* (10 Points)

SeminarSchedulingSystem::schedule(s:Seminar, d : Date)

Pre:

Post:

-----Good Luck-----