COMSATS University Islamabad, Lahore Campus

☐ Sessional-I ☐ Sessional-II ■ Terminal Examination – Spring 2021

Course Title:	Formal Methods			Course	CSE356	Credit Hours:	3(3,0)
Course Instructor/s:	Dr. Farooq Ahmad			Programme	BS Software Engineering		
Semester:	4th	Batch:	FA19-BSE	Section:	A, B	Date:	07th July, 2021
Time Allowed:	**3 hours**			**Maximum Marks:**		**50**	
Student's Name				Reg. No.			

Important Instructions / Guidelines:

- Answer all questions on the exam paper provided to you.
- Do not use the lead pencil.

Question 1: **[Marks: 1+3+3+3=10]**

Formally specify the system in VDM-SL.

Consider a system that records the current mode of an industrial robot, which can either be working, idle or broken.

(a) Declare a type, Mode, for use in the specification.

(b) Define the state of the system in VDM with the state variable *initialMode* of the type Mode including an invariant function. Further, the state includes an initialization function that ensures that the robot is set to idle when the system first comes into existence.

(c) Write specifications for the following operations in VDM-SL:

i. An operation called setMode that accepts and records a value for the mode of the robot.

ii. An operation called getMode that outputs the current mode of the robot.

Question 2: **[Marks: 4+6=10]**

Formally specify the Airport class in Z language.

A system that keeps track of aircraft that are allowed to land at a particular airport. Aircraft must apply for permission to land at the airport before landing. When an aircraft arrives to land at the airport it should only have done so if it had previously been given permission. The invariant property for the system is the landed aircraft are those who have the permission. Assume that the airport can land MAX number of aircraft that could be landed at any one time.

a) Define the state schema in Z.

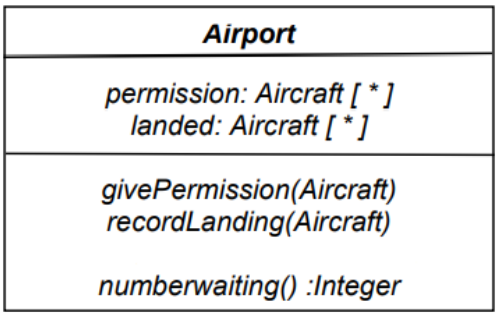
b) Define the operation schemas in Z for the following operations:

givePermission: records the fact that an aircraft has been granted permission to land at the airport.

recordLanding: records an aircraft as having landed at the airport.

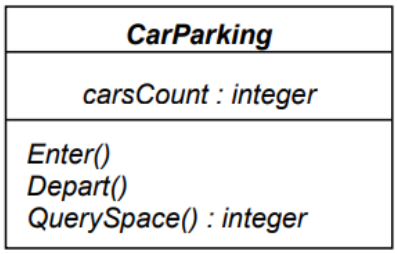
numberWaiting: returns the number of aircrafts granted permission to land but not yet landed.

The UML specification of the Airport class is given below.



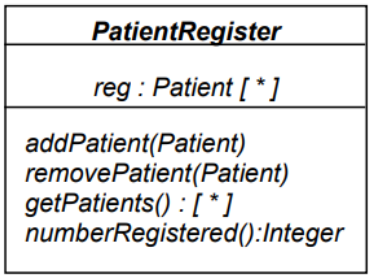
Question 3: **[Marks: 10]**

A mall has a capacity of LIMIT number of spaces for car parking. A count is maintained of the number of cars currently occupying spaces. When a car enters the car park, the count is increased. When a car leaves the car park the count is decreased. The system outputs the number of spaces left in the car park.
Formally specify the CarParking class in VDM-SL.



Question 4: **[Marks: 10]**

Consider a system that registers patients at a doctor's surgery. Assume that the surgery can deal with a maximum of LIMIT number of patients on its register. It will be necessary to add and remove patients from the register. The register must be able to be interrogated so that the list of patients and the number of patients registered can be returned. The UML specification of the PatientRegister class is given below. Formally specify the system in Z.

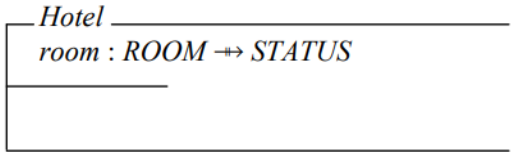


Define the state Schema and operational schemas for PatientRegister along with the constant that number of registered patients must not exceed the LIMIT.

Question 5: **[Marks: 4+3+3=10]**

A hotel management system is used to maintain a record of the current state of the hotel rooms, whether they are occupied or not. The hotel room system comprises a finite number of rooms. We intend that each room is uniquely identified and that each room is either occupied or vacant. Further [ROOM] is declared as the basic type and the status of the room can be defined by using the type:
STATUS::= occupied | vacant

The system state schema is given below:



Specify the operation schemas in Z for the following operations:
AddRoom: the operation adds a new room to the system. A new room is always vacant. We check that the room does not already exist and we add the new room to the system.
OccupyRoom: When a guest arrives, we inform the system that a room is now occupied. We check that the room exists and is not occupied, and we update our room records.
VacantRoom: When a guest departs, we inform the system that the room is now vacant. We check that the room exists and is occupied, and we update our room records.