**EXAMPLE OF EXAM QUESTIONS FOR THE MODULE:**
**OBJECT-ORIENTED METHODS (G52OBJ)**

*Answer QUESTION ONE and TWO OTHERS*

1. A university wishes to increase security in its car park. It has been decided to issue an identity card to all employees. The cards record the employee's name, department and identity number.

A barrier, a card reader and a sensor are placed at the entrance of the car park. The driver inserts the numbered card into the card reader. The card reader checks the card number. If the number is valid, the reader sends a signal to raise the barrier and the vehicle can enter the car park. The sensor sends a signal to the barrier to lower when the vehicle has entered. There is a system at the exit with a barrier which is raised when a car wishes to leave the car park.

When there are no spaces in the car park a sign at the entrance displays "Full" and is only switched off when a vehicle leaves.

Special visitors' cards, which record a number and the current date, also permit access to the car park. Visitors' cards may be sent out in advance or collected from reception. All visitors' cards must be returned to the reception when the visitor leaves the site so that they can be deleted from the list of valid cards.

(a) Draw a UML use case diagram for the university car park system. (6)

(b) Write a step by step use case description for one of the use cases from the use case diagram produced in (a). (9)

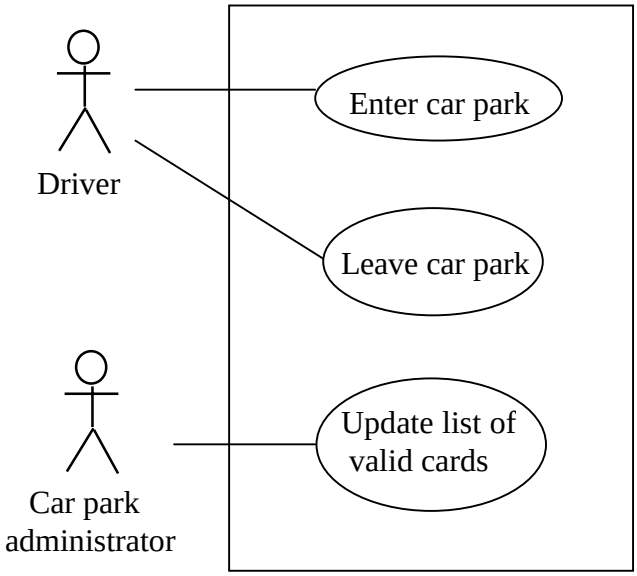
(c) Draw a UML class diagram for the university car park system. The class diagram should represent all of the classes, their attributes and operations, relationships between the classes, multiplicity specifications, and other model elements that you find appropriate. (20)

(d) What is the purpose of the object-oriented analysis? What steps does it consist of? Give a one-sentence description of each of the steps. (10)

(e) Give the definition of the polymorphism. (5)

**Answers**

(a)

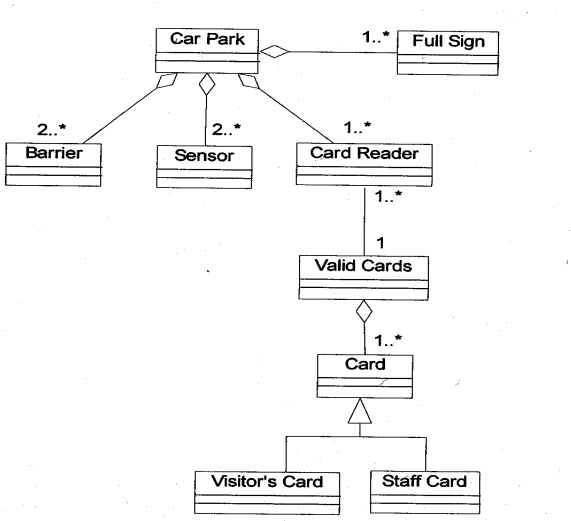



(b) Use case: Enter car park

1. The driver inserts his/her card.
2. The card reader reads the card number.
3. The system checks if the card number is in the list of valid card numbers.
4. The system checks if there is a space.
5. If there is a space and the card is valid the system sends a message to the barrier to lift up.
6. The sensor sends “car passed” message to the system.
7. The sensor sends a message to the barrier to lower.
8. The system decreases the number of available spaces by 1.

Variations.
3.a. Card is not valid
4.a. The car park is full.

(c)




**Attributes and operations**

Car_Park:
Attributes:capacity, spaces
Operations: decr(), incr(), spaces_left(),

Card_Reader
Attributes:
Operations: read_card()

Full_Sign:
Attributes: on
Operations: switch_on(), switch_off(),

Barrier:
Attributes: up
Operations: raise(), lower()

Sensor:
Attributes: car_sensed
Operations: sense_car()

Valid_Cards:
Attributes: list_of_cards
Operations: card_OK(), add_card(), delete_card()

Card:
Attributes: number
Operations: delete()

Visitor'sCard:
Attributes: current_date
Operations: delete()

StaffCard
Attributes: name, department, expiry_date,
Operations: delete()

(d) The purpose of the object-oriented analysis is to show what the desired system must do, not how it will be done. Object-oriented analysis is method that examines requirements from the perspective of the classes and objects.
Steps of the object-oriented analysis:
1. Use case analysis consists of identification of contact partners for the analysis, working material and objects, use cases and actors, describing use cases and clarification of system requirements.
2. Specification of application architecture which denotes the way in which the individual subsystems and components are structured internally.
3. Definition of technical dictionary which should minimise communication problems between developers and users.
4. Development of explorative prototypes to show users and specialists actual aspects of the future system.
5. Construction of CRC cards that show classes, responsibilities and collaborators of the classes.
6. Identifying business classes that describe objects, concepts, places or persons from real business life in a degree of detail that can also be understood by technical departments and managers.
7. Activity modelling to further concretise the requirements and to show how the system is to satisfy these requirements.
8. Component building with the aid of packages. Use cases, business classes, activities from the activity diagrams are classified into components.

(e) Polymorphism denotes the ability to manipulate objects of distinct classes using only knowledge of their common properties without regard for their exact class.
or
Polymorphism is a concept where a single name may denote objects of many different classes that are related by some common superclass.
or
Polymorphism is the ability of objects of different classes to respond to the same message, usually in different ways.

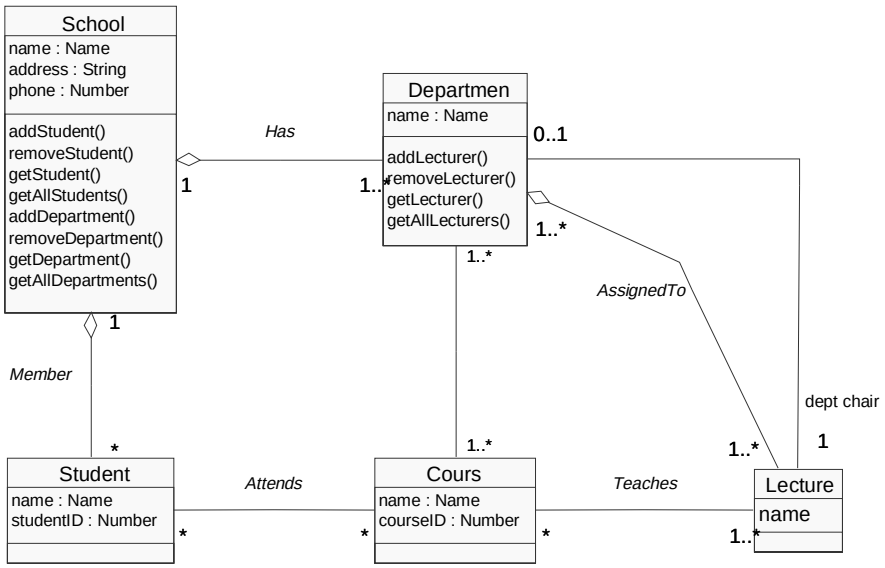
2. You are required to develop an information system for a school. The requirements specification is as follows. A school has one or more departments. Each department has a lecturer who is a department chair. A lecturer can be the chair of no more than one department. Every student may attend any number of courses and every course may have any number of students. For every course there is at least one lecturer. Each lecturer may teach any number of courses.

(a) Draw a class diagram which consists of all classes in your system, their attributes and operations, relationships between the classes, multiplicity specifications, and other model elements that you find appropriate. (15)

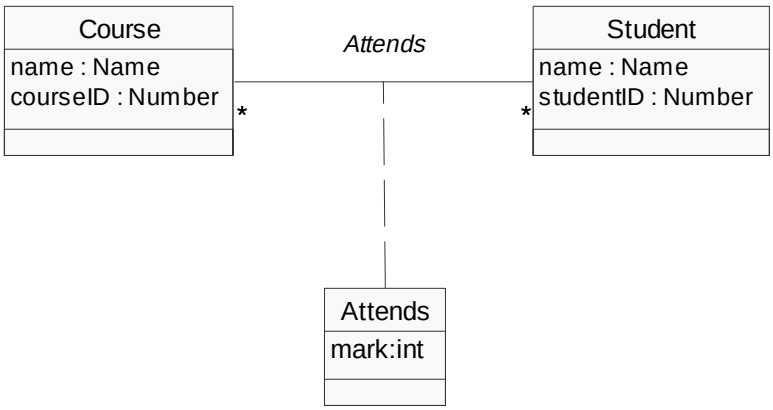
(b) Extend your class diagram so that information on the student's marks on the course is taken into account. Draw the extended part of the class diagram that shows the model element used to record the student's marks. Give a description of the model element that you used.(10)

**Answers**

(a)



(b)




Attributed association is a model element that has both the properties of a class and an association. It is used when relation has properties that cannot be associated neither to one nor to other class. The attributed association is noted like a common class which is called association class. The instances of the association class are the concrete relations between objects of the classes involved in the association. Two objects involved in attributed association may have at most one relation with each other.

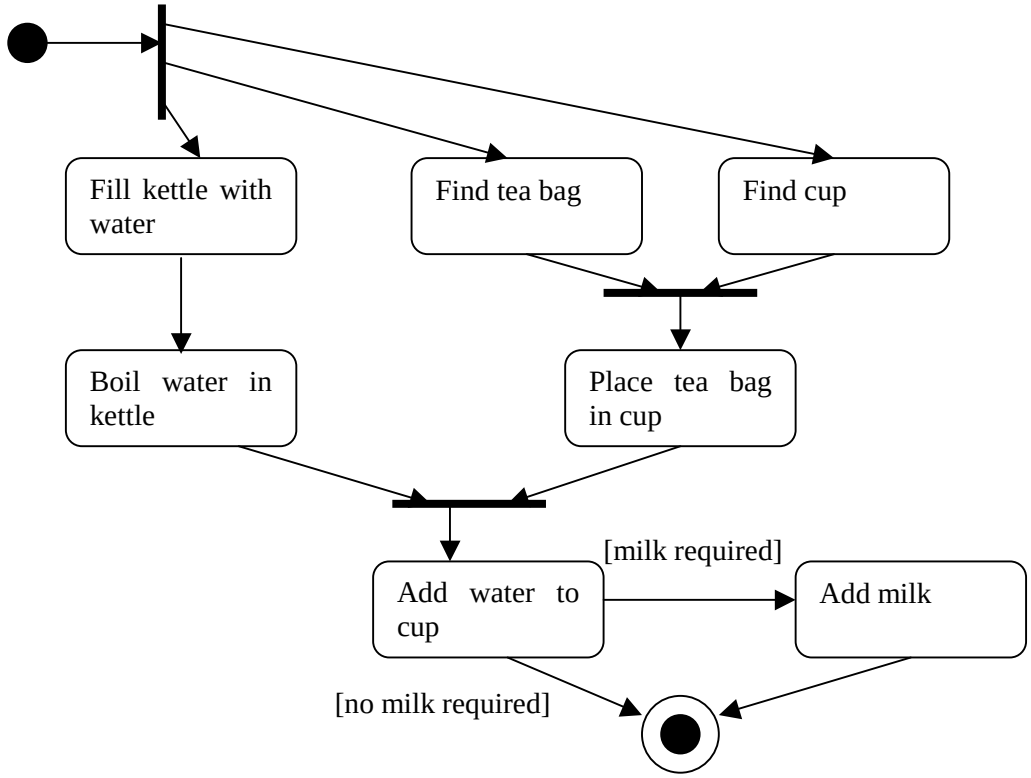
3. (a) Draw an activity diagram that represents the making of a cup of tea. The initial three activities are Fill kettle with water, Find cup and Find tea bag and they may be performed in parallel. When the Find cup and Find tea bag are completed the activity Place tea bag in cup can start. The kettle must have boiled and the tea bag must have been placed in the cup before the activity Add water to cup can begin. If milk is required then activity Add milk should be performed. (9)

(b) Describe the basic idea behind the architectural principle based on Model/View/Controller components and give a short description of each of the components. (8)

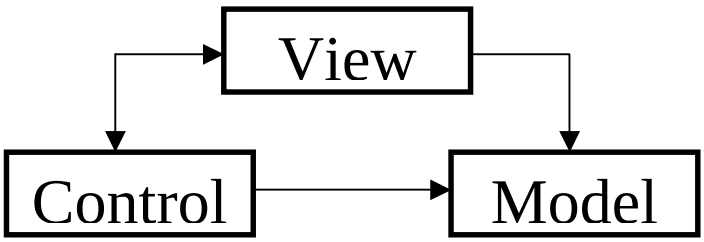
(c) List all behavioural UML diagrams and describe the purpose of each of them. (8)

**Answers**

(a)



(b)




Model holds the actual technical knowledge. View defines the representation of information. Controller controls interaction with the user. View and controller know each other and communicate in both directions. Their relation with the model is one way. Model is independent from other two components.

(c) The UML models that describe the behaviour of the system are:

1. Activity diagrams describe the procedural possibilities of a system with the aid of activities.
2. Collaboration diagrams show a set of interactions between selected objects in a specific situation, focusing on the relations between the objects and their topography.
3. Sequence diagrams show a series of messages exchanged by a selected set of objects with an emphasis on the chronological course of events.
4. State diagrams show a sequence of states an object can assume during its lifetime, together with the stimuli that cause changes of states.

4. Assume that you have to develop a computer system for a university library. The library contains books and each book may have several copies. Only members of the library may borrow books. The system checks whether the potential borrower is a member of the library, and whether there is a reservation on the book. If both checks succeed the system records that the book is on loan. Otherwise it refuses the loan.
A library member may ask to extend the loan of the book. The system then checks whether there is a reservation on the book. If so, the system refuses to extend the loan. Otherwise it records the extension of the loan.

(a) Draw a use case diagram that represents the activities of the members of the library. (5)

(b) Draw a sequence diagram that shows scenario of the use case Borrow copy of book in which the user is permitted to borrow the book, and the system records that the book is on loan. (6)

(c) What is the main difference between collaboration and sequence diagrams? (4)

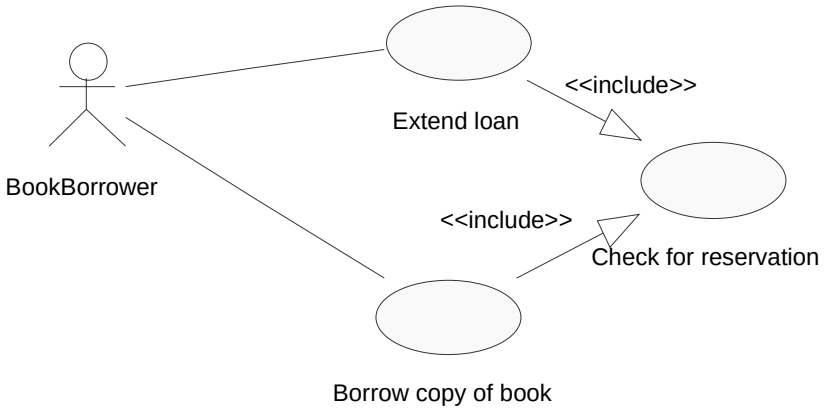
(d) Here is the syntax of a message in collaboration diagrams:

Predecessor_Condition Sequence_Expression Return_Value :-
Message_Name (Argument_List)

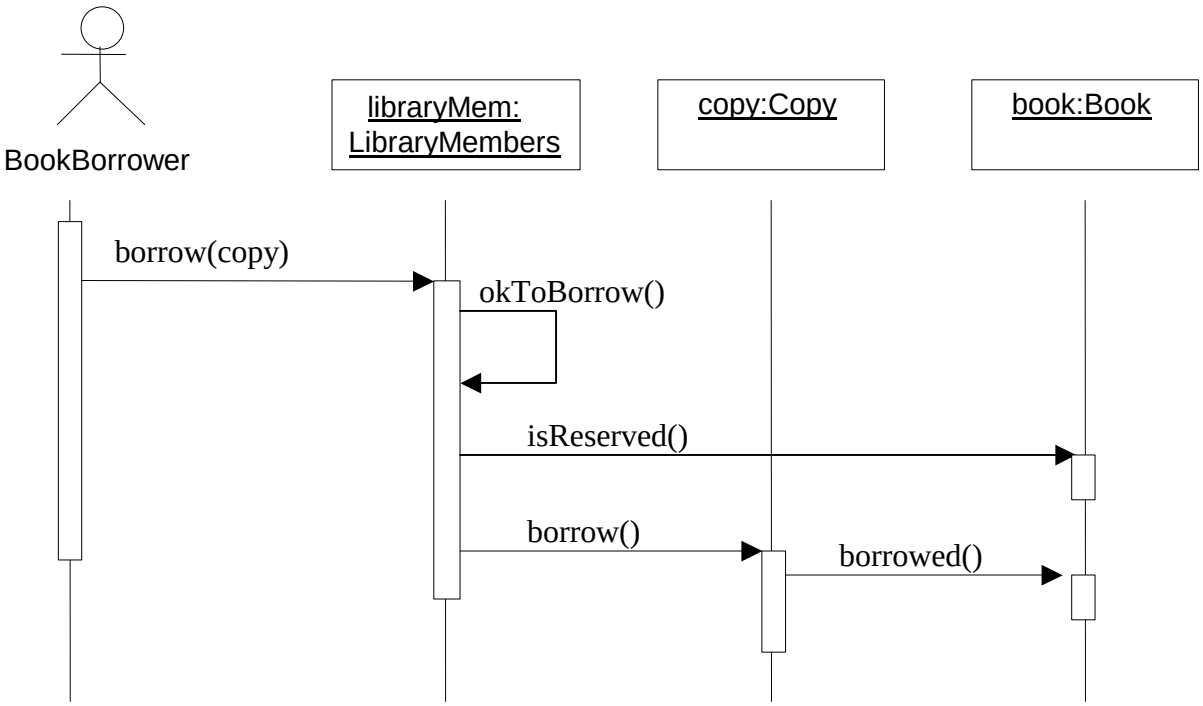
Explain the meaning of each of the elements in the syntax and give an example for each. (10)

**Answers:**

(a)



(b)




(c) The collaboration and the sequence diagrams basically show the same facts. The main difference between these two types of interaction diagrams is that a collaboration diagram emphasises the objects and their cooperation with each other while a sequence diagram shows the chronological sequence of the messages.

(d)
Predecessor_Condition Sequence_Expression Return_Value :-
Message_Name (Parameter_List)

1. Predecessor_Condition is a list of the sequence numbers of other messages that need to have been sent before this message may be sent. The sequence numbers are listed separated by commas and terminated with a /.

example: 1.1, 2.3 /

2. Sequence_Expression shows the sequence of messages numbered in ascending order. It allows for nesting inside messages (a new message is sent within a current message). The message numbers are then separated by a dot.

example: 1.2 follows 1.1

Iterations, or repeated sending of a message, are marked with an asterisk *. To describe the iteration in more detail, for example to specify the number of iterations, a pseudocode can be added in square brackets

example: 1.2.* [i:= 1..n]:

A parallel executions are indicated by ||.

example: 1.2.* || [i:= 1..n]:

A condition can be set which must be satisfied for the message to be sent.

example: [x>5] 1.2.*:

3. Return_Value is a response given by a message. It can be used as a parameter in other messages.

4. Message_Name(Parameter_List) contains the name and the parameters (i.e. actual arguments) of the message. An object which receives the message has an operation which corresponds to the message. The paramteres of the message correspond to the formal arguments in the operation.