CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test - II

Sub:	Object Oriented Modeling and Design							Code:	10CS71
Date:	02 / 11 / 2016	Duration:	90 mins	Max Marks:	50	Sem:	VII	Branch:	CSE/ISE

Answer Any FIVE FULL Questions

		Marks	OBE	
			CO	RBT
1(a)	How does an activity diagram differ from traditional flow chart? Draw an activity diagram for Online ordering system.	[10]	CO 4	L3
2	Explain the steps followed in constructing application interaction model.	[10]	CO2	L2
3(a)	What is a Reuse Plan? Explain the different aspects of re-usability.	[06]	CO1	L2
(b)	Explain any two architectural styles suited for system design.	[04]	CO1	L2
4(a)	Consider shopping in a physical bookstore of super market i) List three actors that are involved in the design of a checkout system. Explain the relevance of each actor. ii) Take the perspective of a customer and list two use cases. Summarize the purpose of each use case within a sentence. iii) Prepare a use case diagram for physical bookstore checkout system. iv) Prepare a normal scenario *for* each use case.	[10]	CO3	L3
5(a)	List and explain the steps involved in organizing of a class design.	[10]	CO 1	L2
6(a)	Explain Sequence models. Draw the Sequence diagrams for an online stock broker.	[10]	CO4	L2
7(a)	Explain how to handle boundary conditions in system design process.	[06]	CO 1	L2
(b)	Explain the conditions for dividing the system into sub modules with appropriate examples.	[04]	CO 1	L2

Course Outcomes		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1:	Explain the steps involved in the OOMD - Requirements, Analysis, Design.(SDLC)	0	0	1	1	0	0	0	0	0	1	0	0
CO2:	Model classes using UML notations covering - Events, Generalization, Association, Links,	2	1	2	2	3	0	0	0	1	0	0	0
CO3:	Model requirements using Use cases and Usecase scenarios.	0	1	0	0	3	0	0	0	0	1	0	0
CO4:	Model Object interaction using state diagram, sequence diagram, Activity Diagram	2	1	2	2	3	0	0	0	0	1	0	0
CO5:	Explain the object oriented methodology to handle legacy systems	2	0	0	0	0	0	0	0	0	0	0	0
CO6:	Explain the following Design Patterns with an example application - Forward-Receiver, Client-Dispatcher-Server, Publisher-Subscriber, Command Process, View handler.	0	0	3	0	0	1	0	0	0	0	0	0

Cognitive level	KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PO1 - *Engineering knowledge*; PO2 - *Problem analysis*; PO3 - *Design/development of solutions*; PO4 - *Conduct investigations of complex problems*; PO5 - *Modern tool usage*; PO6 - *The Engineer and society*; PO7- *Environment and sustainability*; PO8 – *Ethics*; PO9 - *Individual and team work*; PO10 - *Communication*; PO11 - *Project management and finance*; PO12 - *Life-long learning*

CMR INSTITUTE OF TECHNOLOGY

USN

Internal Assessment Test 2 – Nov 2016 - SCHEME OF EVALUATION

Sub:	Object Oriented Modeling and Design			Code:	10CS71
Date: 02/11/2016	Duration: 90 mins	Max Marks: 50	Sem: VII	Branch:	ISE/CSE

Note: : Answer any 5 questions Total marks: 50

1. **How Does an Activity diagram differ from traditional flow chart ? Draw an activity diagram for ordering processing system** **10M**

(Definition of activity diagram and flow chart) 3M

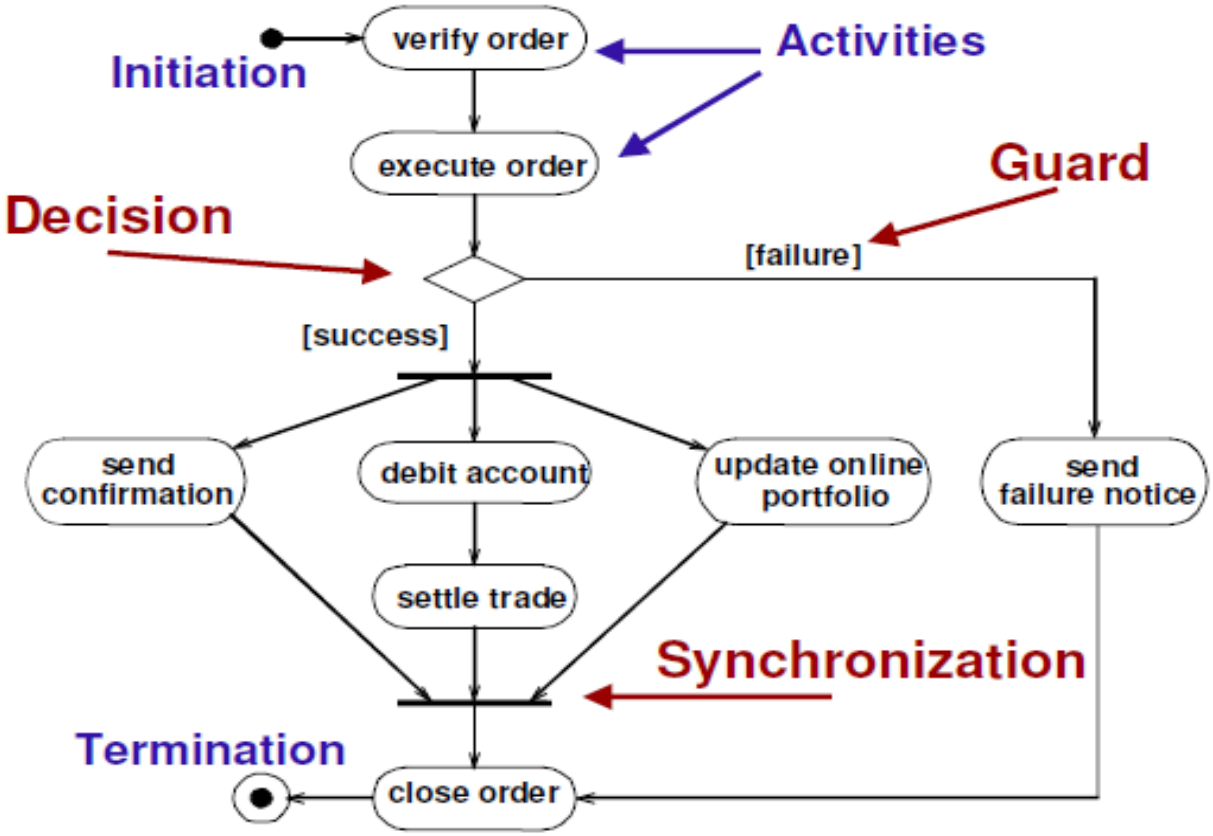
- **Activity diagrams** Activity diagrams and use cases are logical model which describe the business domain's activities without suggesting how they are conduct.
- A diagram that emphasizes the flow of control from activity to activity in an object.
- Similar to the traditional program flowchart.
- Used to provide detail for complex algorithms.

Primary activities and the relationships among the activities in a process.

(difference between Activity diagram and Flowchart) 2M

Hence they can be regarded as a form of flowchart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

Activity diagram for ordering processing system **5M**



2. Explain the steps followed in constructing application interactive model **10M**
(Ten points with explanation each carries 1 M)

Steps to construct model

1. Determine the system boundary
2. Find actors
3. Find use cases
4. Find initial and final events
5. Prepare normal scenarios
6. Add variation and exception scenarios
7. Find external events
8. Prepare activity diagrams for complex use cases.
9. Organize actors and use cases
10. Check against the domain class model **10*1=10**

3 a. What is a Reuse Plan? Explain the different aspects of re-usability. 6 M

Two aspects of reuse:

- Using existing things
- Creating reusable new things

(Three things for reusing) 3*2=6M

•Reusable things include:

- Libraries
- Frameworks
- Patterns

1. Reusable Libraries **2M**

A library is a collection of classes that are useful in many contexts.

Qualities of "Good" class libraries:

- *Coherence* – well focused themes
- *Completeness* – provide complete behaviour
- *Consistency* - polymorphic operations should have consistent names and signatures across classes
- *Efficiency* – provide alternative implementations of algorithms.

2. Reusable Frameworks **2M**

- A framework is a skeletal structure of a program that must be elaborated to build a complete application.
- Frameworks class libraries are typically application specific and not suitable for general use.

3. **Reusable Patterns** **2M**

- A pattern is a proven solution to a general problem.
- There are patterns for analysis, architecture, design, and implementation.
- A pattern is more likely to be correct and robust than an untested, custom solution.

- Patterns are prototypical model fragments that distill some of the knowledge of experts.

Pattern vs. Framework

- A pattern is typically a small number of classes and relationships.
- A framework is much broader in scope and covers an entire subsystem or application.

3 b. Explain any two architectural styles suited for system design. 4M

Several prototypical architectural styles are common in existing system.

Some kinds of systems:

1. Batch transformation
2. Continuous transformation
3. Interactive interface
4. Dynamic simulation
5. Real-time system
6. Transaction manager

(out of 6 styles explain any 2 styles each carries 2 Marks) 2*2=4M

4. Consider shopping in a physical bookstore of super market 10M

i) List three actors that are involved in the design of a checkout system. Explain the relevance of each actor.

ii) Take the perspective of a customer and list two use cases. Summarize the purpose of each use case within a sentence.

iii) **Prepare a use case diagram for physical bookstore checkout system.**

iv) **Prepare a normal scenario *for* each use case.**

(each sub question carries 2.5 M) 4*2.5=10M

7.1 Here are answers for a physical bookstore.

a. Some actors are:

- **Customer**. A person who initiates the purchase of an item.
- **Cashier**. An employee who is authorized to check out purchases at a cash register.
- **Payment verifier**. The remote system that approves use of a credit or debit card.

b. Some use cases are:

- **Purchase items**. A customer brings one or more items to the checkout register and pays for the items.
- **Return items**. The customer brings back items that were previously purchased and gets a refund.

c. Figure A7.1 shows a use case diagram.

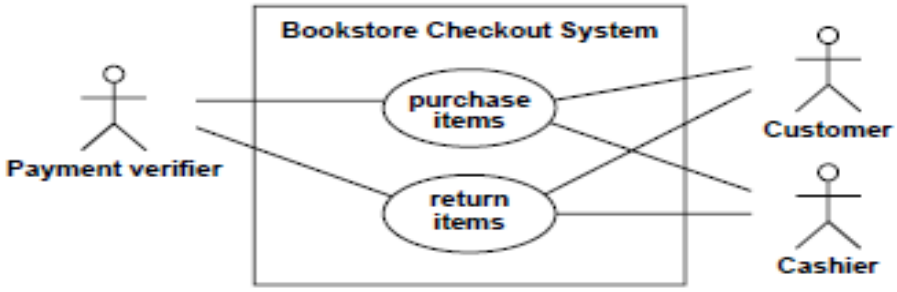


Figure A7.1 Use case diagram for a physical bookstore checkout system

d. Here is a normal scenario for each use case. There are many possible answers. [Instructor's note: You may wish to give the students the answers to Exercise 7.1 parts a-c.]

- **Purchase items.**
  Customer brings items to the counter.
  Cashier scans each customer item.
  Cashier totals order, including tax.
  Cashier requests form of payment.
  Customer gives a credit card.
  Cashier scans card.
  Checkout system communicates scan data to verifier.
  Verifier reports that credit card payment is acceptable.
  Customer signs credit card slip.
- **Return items.**
  Customer brings purchased item to the counter.
  Customer has receipt from earlier purchase.
  Cashier notes that payment was in cash.
  Cashier accepts items and gives customer a cash refund.

e. Here is an exception scenario for each use case. There are many possible answers. [Instructor's note: You may wish to give the students the answers to Exercise 7.1 parts a-c.]

- **Purchase items.**
  Customer brings items to the counter.
  Cashier scans each customer item.
  An item misscans and cashier goes to item display to get the item price.
- **Return items.**
  Customer brings purchased item to the counter.
  Customer has no receipt from earlier purchase.
  Customer is given a credit slip, but no refund.

5. List and explain the steps involved in organizing of a class design. **10M**

- The analysis phase determines *what the implementation must do*
- The system design phase *determines the plan of attack*
- The purpose of the class design is to complete the *definitions of the classes and associations and choose algorithms for operations*. **1M**

(Nine points with explanation each carries 1 M) **9*1=9M**

Steps to involved in organising class model

1. Bridging the gap
2. Realizing Use Cases
3. Designing Algorithms
4. Recurring Downward
5. Refactoring
6. Design Optimization
7. Reification of Behaviour
8. Adjustment of Inheritance
9. Organizing a Class Design

6. Explain Sequence models. Draw the Sequence diagrams for an online stock broker. 10M

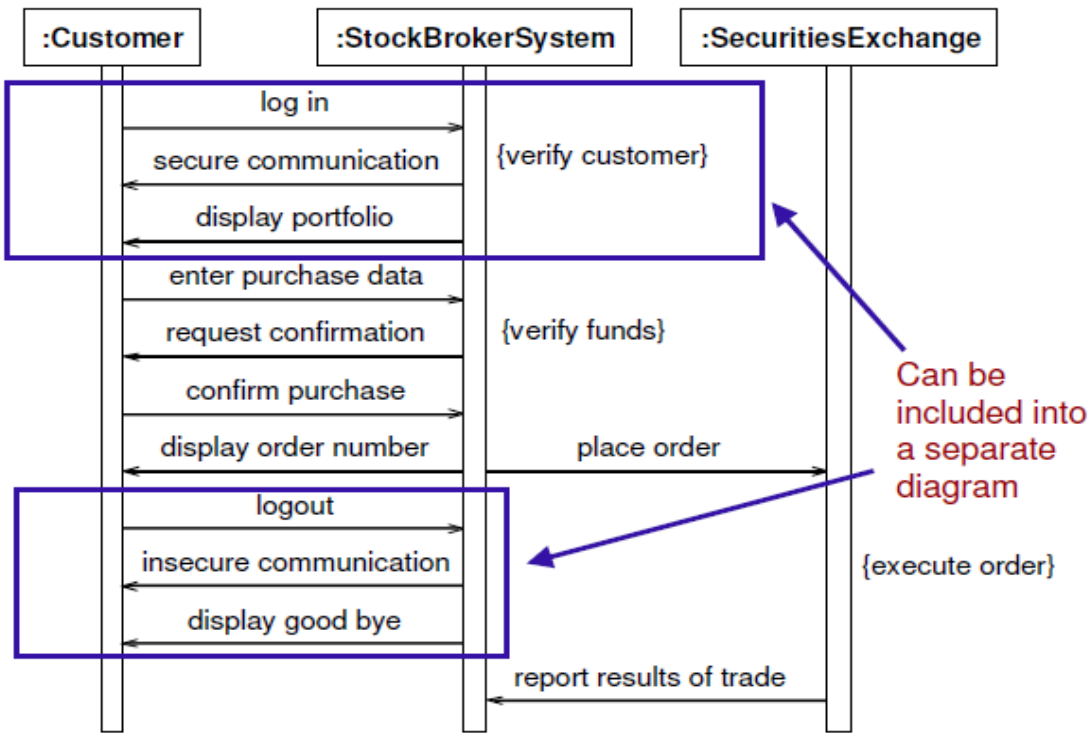
(Definition of Sequence models) 2M

- A sequence diagram shows the participants in an interaction and the sequence of messages among them.
- A sequence diagram shows the interaction of a system with its actors to perform all or part of a use case.
- Each use case requires one or more sequence diagrams to describe its behaviour.

- Sequence diagrams, also known as event diagrams or event scenarios, illustrate how processes interact with each other by showing calls between different objects in a sequence.

These diagrams have two dimensions:

- The vertical lines show the sequence of messages and calls in chronological order
- Horizontal elements show object instances where the messages are relayed.



Sequence diagrams for an online stock broker with explanation 5+3 M

7a. Explain how to handle boundary conditions in system design process **6M**

- Most of system design is concerned with steady-state behaviour, but boundary conditions are also important
- Boundary conditions are

1. Initialization
2. Termination
3. Failure

(Each Boundary with explanation carries 2 M) **3*2=6M**

7b. Explain the conditions for dividing the system into sub modules with appropriate examples. **4M**

- A subsystem is a group of classes, associations, operations, events, and constrains.
- A subsystem is usually identified by the services it provides.
- Each subsystem has a well-defined interface to the rest of the system.

- **The relation between two subsystems can be**
  a. **Client-server relationship**
  b. **Peer-to-peer relationship**

- Subsystems is organized as a sequence of

– Horizontal layers,
– Vertical partitions, or
– Combination of layers and partitions.

Layered System with explanation **2M**

- Each built in terms of the ones below it.
- The objects in each layer can be independent.
  E.g- A client-server relationship
- Problem statement specifies only the top and bottom layers:
  – The top is the desired system.
  – The bottom is the available resources.
- The intermediate layers is than introduced.

Partitioned system explanation **2M**

- Vertically divided into several subsystems
- Independent or weakly coupled
- Each providing one kind of service.
  - E.g. A computer operating system includes
    – File system
    – Process control
    – Virtual memory management
    – Device control