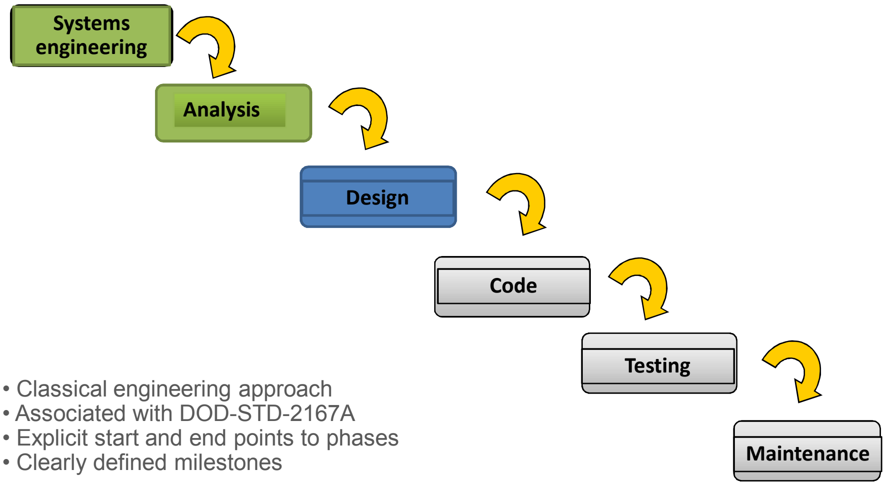
CSE 4214
Introduction to Software Engineering

11

Design

Collaboration & Responsibilities

"Classic" Waterfall Model[1]



- Classical engineering approach
- Associated with DOD-STD-2167A
- Explicit start and end points to phases
- Clearly defined milestones

Software Engineering Process

This Class

- System Engineering:
  - *Understand the context of the software in the overall system where it will be used.*
- Analysis:
  - *purpose is to understand the problem for which software solution is going to be provided.*
  - *It is done through requirements gathering and analysis, developing models, etc.*
  - *This is also a time to come up with a plan to validate the software.*
- Design = Preliminary (Architectural) + Detailed
  - *Purpose is to provide a solution to the problem specified during analysis.*
  - *It is done by developing architecture and dynamic models for the software system. Also, algorithms & data structures are finalized.*
  - *Based on the design information specific tests are developed.*
- Code:
  - *Develop source code from the design for the software.*

Process Steps & Documentation

This Class

- System Engineering:
  - Concept of operation (ConOp)
- Analysis:
  - Software requirements specification (SRS)
  - Human computer interface (HCI)
  - Acceptance test plan (ATP)
- Design = Preliminary (Architectural) + Detailed
  - Architectural design specification (ADS)
  - Detailed design description (DDD)
  - Test cases (TC)
- Code:
  - Source & executable code

Review

UML Models

- Analysis models
  - Use case
  - Class diagram // with classes from the problem domain
  - Sequence diagram // System level; user-system interactions
- Design models
  - Collaboration diagram
  - Class diagram // with design classes
  - Sequence diagram // includes interactions between classes within the system; optional

Design Models

- Static models
- Dynamic models

LibSys Design

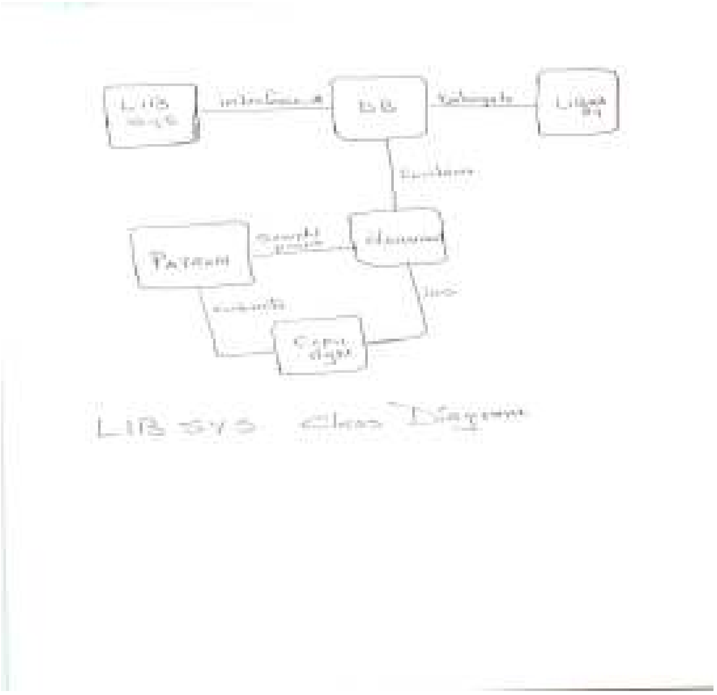
- Classes in the initial class diagram

  (from requirements analysis)

  – LibSys Database (DB) Library

  – Document (Doc) Patron Copy_right

Object model LIBSYS



Collaboration Model

Collaboration Diagram

- A dynamic UML model
- Shows how a system function is performed inside the system
- Shows the objects working together to accomplish a function of the system
- There is a collaboration diagram for each system function

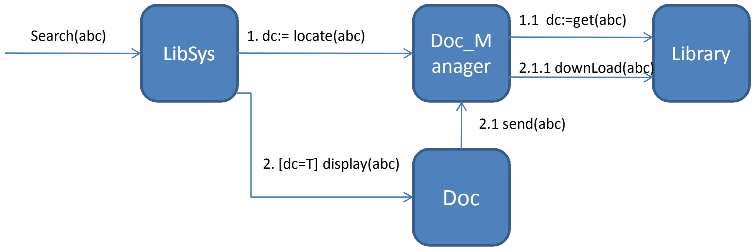
Collaboration Diagram

elements / notations

- External event
- Controller class & collaborators
- Message passing
- Message passing sequence
- Conditional message passing

Collaboration Diagram

Search for document



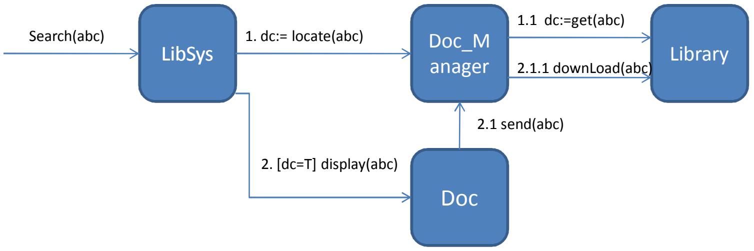
Collaboration Diagram

significance

- Shows subset of objects in the system, which collaborate to do each function of the system
- All the classes/objects in the collaboration diagrams will be implemented in the software
  - they are design classes/objects
- Helps to identify operations (methods) in each class
- Provides an algorithm for the system function

Collaboration Diagram

Search for document



Patron requests **LibSys** to search for document *abc*

LibSys sends message <u>locate</u> to **Doc_Manager** & set *dc* ← True if located

Doc_Manager sends message <u>get</u> to **Library**

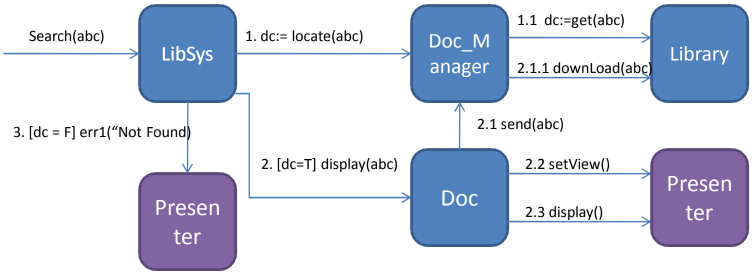
If (document found): **LibSys** sends message <u>display</u> to **Doc** to display *abc*

Doc sends message send to **Doc_Manager**

Doc_Manager sends message <u>downLoad</u> to **Library**

Collaboration Diagram

model & presentation



Patron requests **LibSys** to search for document *abc*

LibSys sends message <u>locate</u> to **Doc_Manager** & set *dc* ← True if located

Doc_Manager sends message <u>get</u> to **Library**

If (document found): **LibSys** sends message <u>display</u> to **Doc** to display *abc*

Doc sends message send to **Doc_Manager**

Doc_Manager sends message <u>downLoad</u> to **Library**

Doc sends messages to **Presenter**

IF (document NOT found): **LibSys** send message <u>err1</u> to **Presenter**

Collaboration Diagram

Print document

- Algorithm

(1) model

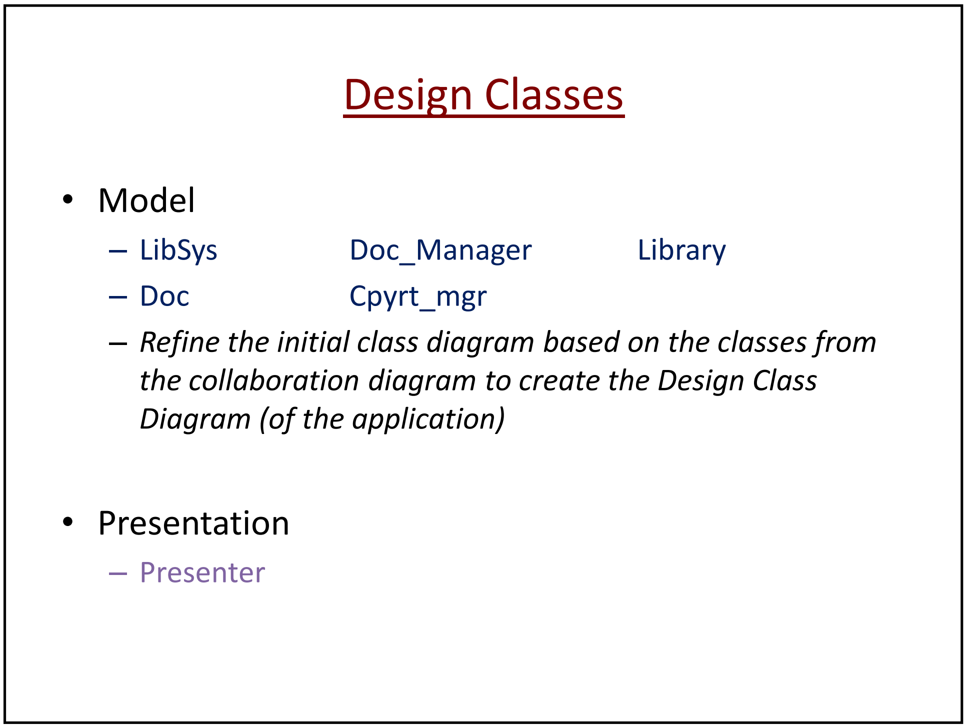
```
Patron requests LibSys to print the document with the message printDoc()
  IF (document displayed) LibSys sends message checkCprt to Cpyrt_Manager // checking copy right
  IF (copy right accepted) LibSys sends message to Doc to print abc
```

(2) model & presentation

```
Patron requests LibSys to print the document with the message printDoc()
  IF (document displayed) LibSys sends message checkCprt to Cpyrt_Manager // checking copy right
      Cpyrt_Manager sends message to Presenter to display copy right form and prompt Patron
  IF (copy right accepted) LibSys sends message to Doc to print abc
      Doc sends message to Presenter to print the document
```

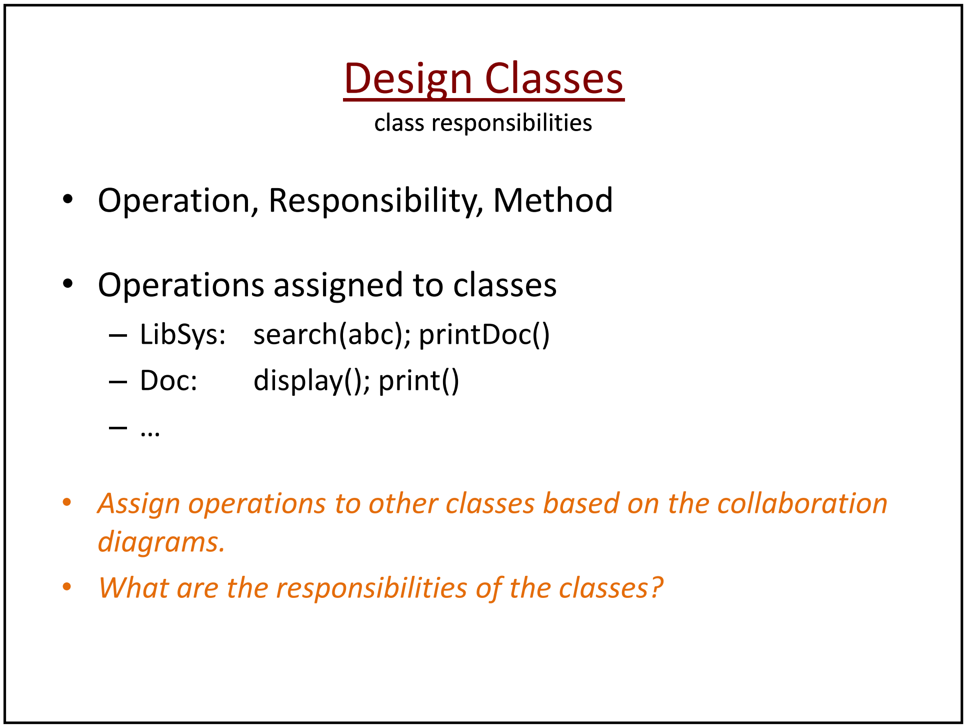
Collaboration Diagram & Design Class Diagram

- Refine the class diagram from analysis phase using classes from collaboration diagram
- remove all classes that are not in any of the collaboration diagram
- add classes that are in CD but not in the class diagram
- The resulting class diagram is design class diagram
- All the classes in the design class diagram must be implemented in the system



Design Classes

- Model
  - LibSys Doc_Manager Library
  - Doc Cpyrt_mgr
  - *Refine the initial class diagram based on the classes from the collaboration diagram to create the Design Class Diagram (of the application)*
- Presentation
  - Presenter



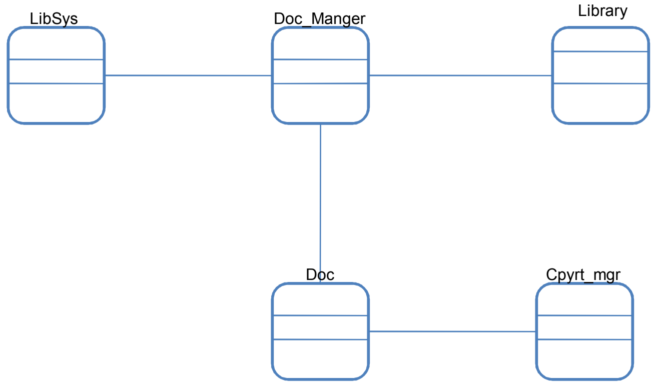
Design Classes

class responsibilities

- Operation, Responsibility, Method
- Operations assigned to classes
  - LibSys: search(abc); printDoc()
  - Doc: display(); print()
  - …
- *Assign operations to other classes based on the collaboration diagrams.*
- *What are the responsibilities of the classes?*

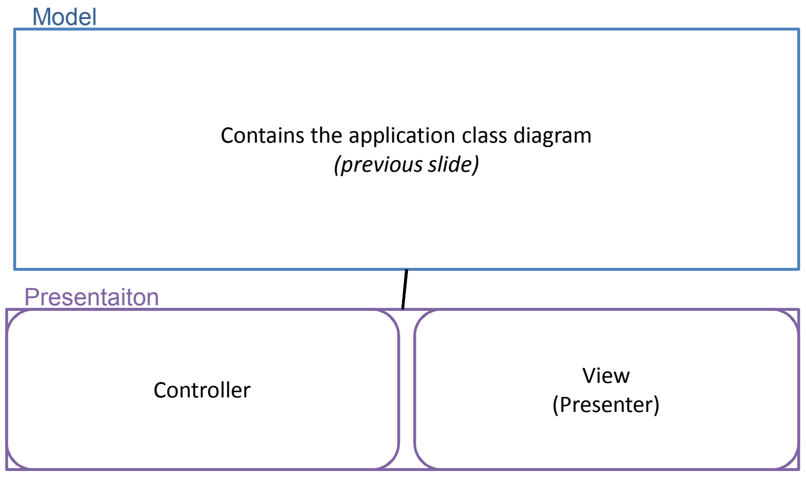
Design Class Diagram

Application



Package Design Classes

Application & Presentation



Model

Contains the application class diagram
(previous slide)

Presentaiton

Controller

View
(Presenter)

Do you know

- Discuss a model. How are models used?
- What are the different types of software models?
  - How are they different?
- What are the UML models covered in the class?
  - When (at which stage) is each model used?
  - Discuss the notations used for each model.
  - Describe each model and its use in software development.
  - Develop each model for a given problem.
- CRC method to develop interaction models

Do you know

- UML Models
  - What is the difference between domain classes and design classes.
  - How does a collaboration diagram help a designer?
  - Assign operations to classes from collaboration diagrams
  - Identify design classes from collaboration diagrams.
  - Discuss the concepts: class responsibility, operation, and method
  - Develop a design class diagram from collaboration diagrams.
- Describe a software development process based on UML models discussed in the class.
- Map collaboration diagram to sequence diagram
- Organize the classes into packages