

# MIDTERM SOLUTION

## Review

Miami University Software Technology & Analysis Group (MUSTANG)  
Computer Science & Software Engineering  
Miami University, Oxford, Ohio, USA

## PART 1 – QUESTION 1

- Suppose your team's **velocity** is **0.6**: Your Iteration length is **10 days** and you have **6 people** on your project. **How much actual work**, measured in person-days, can your team complete? (Show your calculation.)
- **Solution:**  $10 \text{ days} * 6 \text{ people} * 0.6 = 36 \text{ people-days}$

## PART 1 – QUESTION 2

- If Tasks are useful because they are smaller than User Stories, **why not just skip User Stories and write-up Tasks?** (Explain.)
- **Solution:** User stories are written in the customers language to facilitate communication, whereas Tasks are written in technical language, which is not familiar to the customer.

## PART 1 – QUESTION 3

- Translate the following requirements into user stories (omit estimates and priorities): ***I want a chat client in which I can send messages to a given buddy, as well as to all users who have the variable openFlag set to TRUE.***
- **Solution:**
  - Have a feature to set the user's variable to true or false.
  - Allow users to receive messages from non-buddies.
  - Create the ability to send messages to users/buddies (touch of task language)
  - The user should have the ability to list all the users who want to chat
  - Users have the ability to create profiles (buddy list)
  - User should be able to send messages to their buddies and anyone else...

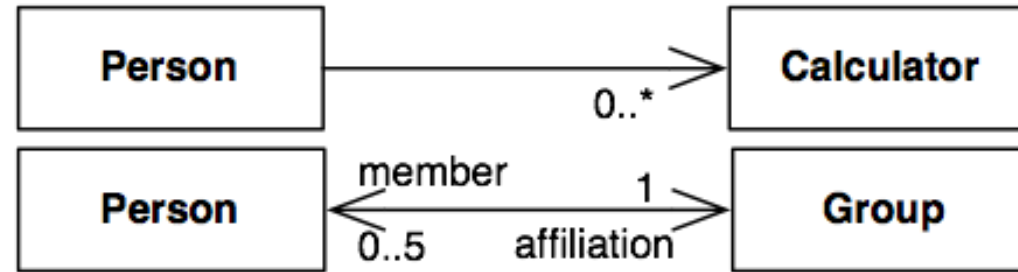
## PART 1 – QUESTION 4

- When new features are added to your product but the deadline is still the same, the re-planning process to consider these features includes the following steps?
- **Solution:**
  - Estimate the new features
  - Have your customer prioritize the new features.
  - Rework your iteration plan
  - Check your project deadline

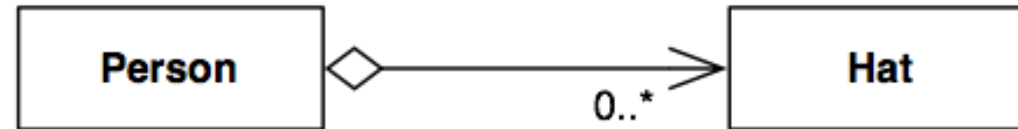
# PART 1 – QUESTION 10

## Association

(Bi-Directional)



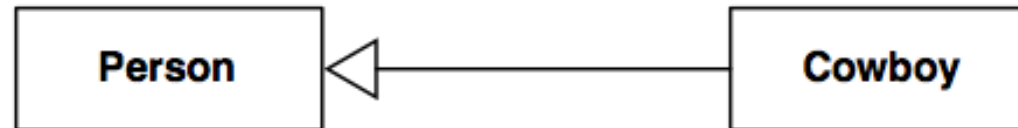
## Aggregation



## Composition



## Inheritance



## PART 2

- The CSE department needs an automated system to help in assigning faculty to courses and courses to appropriate rooms. This system will maintain a list of courses (course number, name) and a notation about the kind of room needed for that course – computer lab, lecture room, seminar room. Each semester, each professor uses the GUI to indicate which courses that they prefer to teach the next semester in priority order. When this process is complete, the department's chair initiates the scheduling process. He or she must first indicate for each faculty member how many courses that they will teach. (For example, some faculty may be mentoring several graduate students, so they may teach fewer courses.) The system then creates a tentative list of teaching assignments that is presented to the chair. He or she then reviews the course assignments for each faculty member. He or she may OK that assignment, or can indicate that he or she doesn't want a particular course assigned to that faculty member. If all the assignments are approved, the process is finished. However, if there is even one course assignment that is rejected, then the system has to redo its process since this may affect the entire schedule. Then the system creates a time schedule for courses that attempt to match faculty time requests.
- This system needs to be moderately reliable. If it fails during the scheduling process, there is normally some slack time. However, this failure should be fixed and the system restored to operation within a week. This system needs to be accessible over the web. The interface should be graphical, and should minimize typing as much as possible.

## PART 2 – QUESTION 1

- Who are the actors in this problem statement?
- **Solution:**
  - Department Chair
  - Faculty member
  - Course and Room Database (This could be an internal object rather than an external actor.)



## PART 2 – QUESTION 2

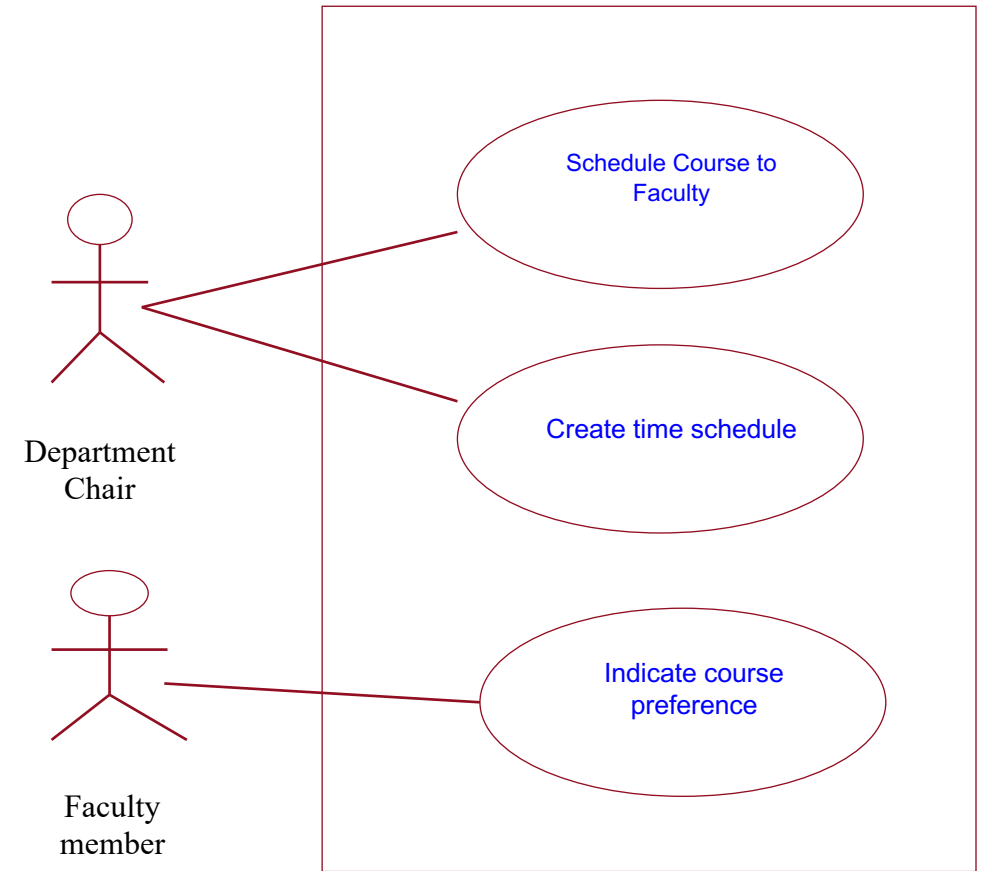
- What is one non-functional requirement that is specified in this problem statement?
- **Solution:**
  - This system needs to be moderately reliable. If it fails during the scheduling process, there is normally some slack time. However, it should be fixed within a week.
  - This system needs to be accessible over the web.

## PART 2 – QUESTION 3

- One of the use cases in this problem is the “**Schedule Course to Faculty.**” This use case is the one in which faculty are actually given course assignments. What are names of two other use cases?
- **Solution:**
  - Indicate course preference for faculty member.
  - Create time schedule
  - (**Others are possible**)

## PART 2 – QUESTION 4

- Draw the one use case diagram that captures all three of these uses cases:  
“**Schedule Course to Faculty**”  
and your **two** use cases.

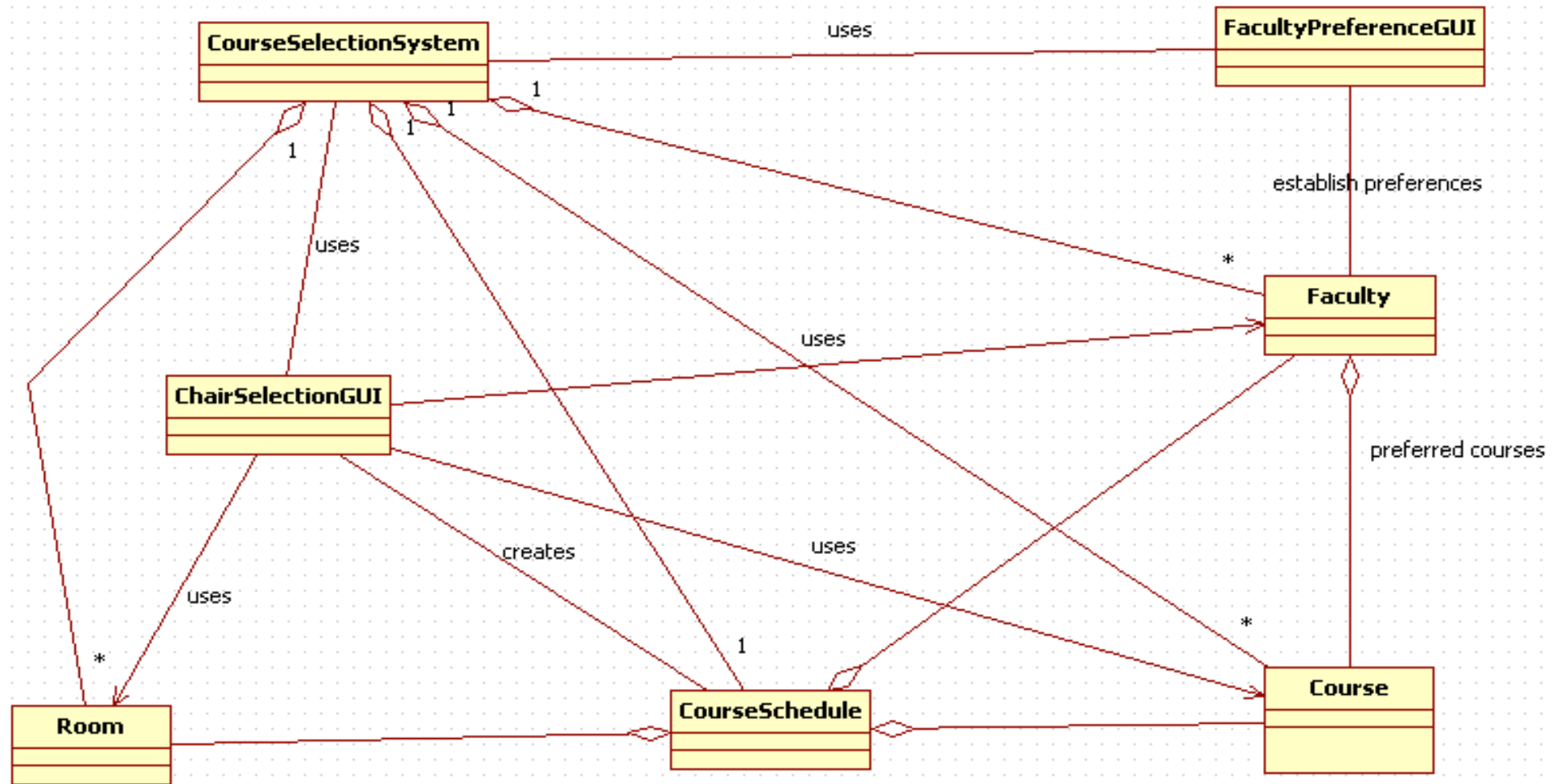


## PART 2 – QUESTION 5

- List the names of the conceptual classes for this system.
- **Solution:**
  - Course, Faculty, Room, CourseSchedule

## PART 2 – QUESTION 6

- Draw a UML **class diagram** that models all the requirements and shows the conceptual classes, key attributes, and their relationships.



## PART 2 – QUESTION 7

- For the use case “**Schedule Course to Faculty**” draw the system sequence diagram.

