

# CSC 113 Project – Second Semester 1446

## Phase 1

### Due date: 6/03/2025

---

The goal of this project is to provide you with hands-on experience with creating your own Java program from design to implementation. This project is to be completed in groups of **2 to 3 students** at most.

**Project Title:** Java project phase 1

**Project Description:** In this project, you will create a Java program using all the skills and topics learned during the course so far. You should come up with an idea for a system that supports user interaction (That is: menu-style input-output) and provides several services for the user.

Some examples of Java projects:

- Hotel reservation system
- Library management system
- Online restaurant ordering system
- Bank management system

You can be as creative as you want as long as you follow the project requirements. Remember, programming is not typing Java code, it involves planning and careful designing.

#### Project Requirements:

- Your system should have **at least five different classes(not including the test class with the main method)**.
- Your system should have **at least two relation** (Composition and Aggregation , it can be with same classes or different classes) , based on your design you should select the appropriate relation.
- You should have **at least four subclasses (inheritance relation)**, you must select the appropriate classes for this relation ( the inheritance must be one level and two level ).
- Your system must support polymorphism with **at least one polymorphic method**.
- At least one class must be an (**Abstract class**).
- There must be **at least two arrays of objects** with methods that support: **add, remove, and search for objects**.
- All the classes must have the necessary attributes and methods to work properly.



### Submission Guidelines:

- You are expected to work in teams, each team can be from 2 to 3 students.
- You will submit a report that includes the following:
  - A cover sheet including the names and IDs of your team members and the division of work among them.
  - Introduction about your program (idea, services provided, etc)
  - Full UML diagram showing the different relations between all your classes.
  - Implementation and design details describing all methods. In addition to submitting your source code (compressed in ZIP format).
  - Screenshot of your sample run .
- All submissions, should be made through LMS (one submission per team) refer to the note for further details.
- The submitted work should be your own work. Any case of unethical conduct will result in an F in the class.

### Note:

- Your code should be written with: proper indentation, comments, and proper naming of the variables, methods, and classes. All of these points will be graded.
- The system you design in this phase will be used in phase 2 as well, so make sure you design it properly.
- Each team should have a **leader**, the leader responsibility is to submit the work on behalf of the team.

---

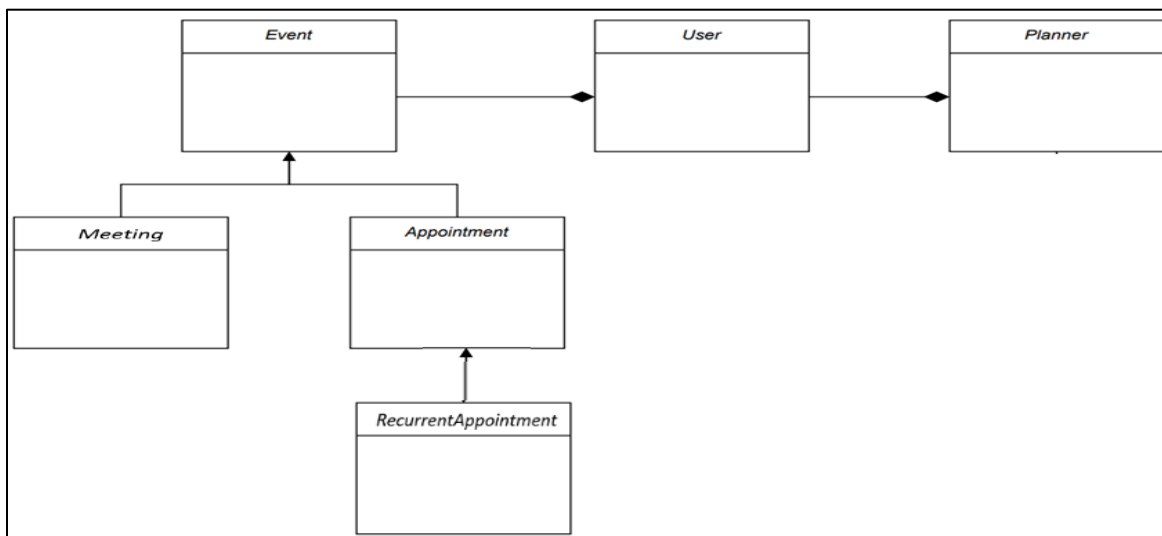
### Example of a Java project:

**Project idea:** An event planner.

**Project introduction:** a simple planning application to keep track of events and manage them. It can be used by more than one user, each with their separate data that can be accessed after a user logs in.

The program has a: Planner, User, Event: the event can be an Appointment or a RecurrentEvent. There is a separate class Application to test the Planner.

**UML:**



**Note:** this is an abstract UML diagram, you need to add more details such as attributes, methods, multiplicity (this is only to give you an idea of what is expected).

**Classes and method-headers:**

**// add the classes and method-headers with a screenshot of your sample run**

**Example:**

Class User:

Methods:

- public boolean addEvent(Event e)

adds event *e* to the User's events if there's space, and returns true if added or false otherwise.

**Sample Run:**

```
***** event planner *****
Welcome to event planner program , what do you need today ? (Enter your menu option)
1. Add event
2. Find event
3. Cancel event
4. Display all events for that user
5. Exit
your choice is : 1
----- Add event -----
```