

CS304 Assignment 1 Solution (Fall 2026)

Object Oriented Programming: Showbiz Film Production System

Total Marks: 20 | Due Date: 5 November 2026 | Lectures 1 to 6

Scenario Summary

A film production company plans to produce a new showbiz film titled The AI and Challenges. The company appoints a producer who manages the overall production, hires the staff, arranges equipment, allocates resources and prepares the shooting schedule. The producer hires a scriptwriter and crew members (a director, an editor and actors). The scriptwriter develops the story and prepares a screenplay made of multiple scenes, and every scene contains dialogues. The director defines the roles, assigns them to suitable actors and guides the actors during rehearsals and filming. After filming, the editor works on the recorded footage to produce the final version of the film.

Step 1: Identified Objects

Every important noun in the scenario that has its own data and behavior becomes a class:

Class	Why it is an object
Company	The film production company that appoints the producer.
Producer	Manages the whole production and hires the staff.
ScriptWriter	Develops the story and prepares the screenplay.
ScreenPlay	The written screenplay made of scenes.
Scene	One scene of the screenplay.
Dialogue	A dialogue spoken inside a scene.
CrewMember	Common base class for all crew members.
Director	Defines roles, assigns them and guides the actors.
Editor	Edits the recorded footage into the final film.
Role	A role required for the film.
Actor	Performs an assigned role.

Step 2: Attributes and Functions

Attributes are kept private (shown with a minus sign) and functions are public (shown with a plus sign). This is encapsulation. Producer, Director and Actor each carry 2 attributes and 2 functions, as required.

Class	Attributes	Functions
Company	companyName, location	appointProducer(), produceFilm()
Producer	producerId, name	hireCrew(), prepareSchedule()
ScriptWriter	storyTitle	writeStory(), createScreenPlay()
ScreenPlay	title	addScene()
Scene	sceneNumber	addDialogue()
Dialogue	text	speak()
CrewMember	crewId, name	work()
Director	directorId, name	defineRole(), assignRole()
Editor	editorId	editFootage(), finalizeFilm()
Role	roleName	assignTo()
Actor	actorId, name	rehearse(), performRole()

Step 3: Relationships Among Objects

Relationship	Type	Reason
Company to Producer	Aggregation	The company appoints a producer, but a producer can exist without this company, so it is a has-a relationship, not ownership.
Producer to CrewMember	Association	The producer hires the crew members.
Producer to ScriptWriter	Association	The producer hires the scriptwriter.
ScriptWriter to ScreenPlay	Association	The scriptwriter creates the screenplay.
ScreenPlay to Scene	Composition	A screenplay is made of scenes. Scenes cannot exist without the screenplay, so the diamond is filled.
Scene to Dialogue	Composition	A scene is made of dialogues. Dialogues belong to their scene.
CrewMember to Director	Inheritance	A director is a crew member and inherits crewId, name and work().
CrewMember to Editor	Inheritance	An editor is a crew member and inherits the common crew data.
CrewMember to Actor	Inheritance	An actor is a crew member, as stated in the scenario, and

		inherits the common crew data.
Director to Role	Association	The director defines the roles required for the film.
Role to Actor	Association	A role is assigned to a suitable actor, who performs it.
Director to Actor	Association	The director guides the actors during rehearsals and filming.

Step 4: Object Oriented Concepts Applied

Encapsulation: attributes are private and accessed only through the public functions of the class, which is why attributes carry a minus sign and functions carry a plus sign in the diagram.

Inheritance: Director, Editor and Actor inherit from CrewMember, so they reuse crewId, name and work() instead of repeating them. This is generalization: the common crew features sit in one base class.

Association: plain working relationships such as Producer hires CrewMember, Director defines Role and Director guides Actor.

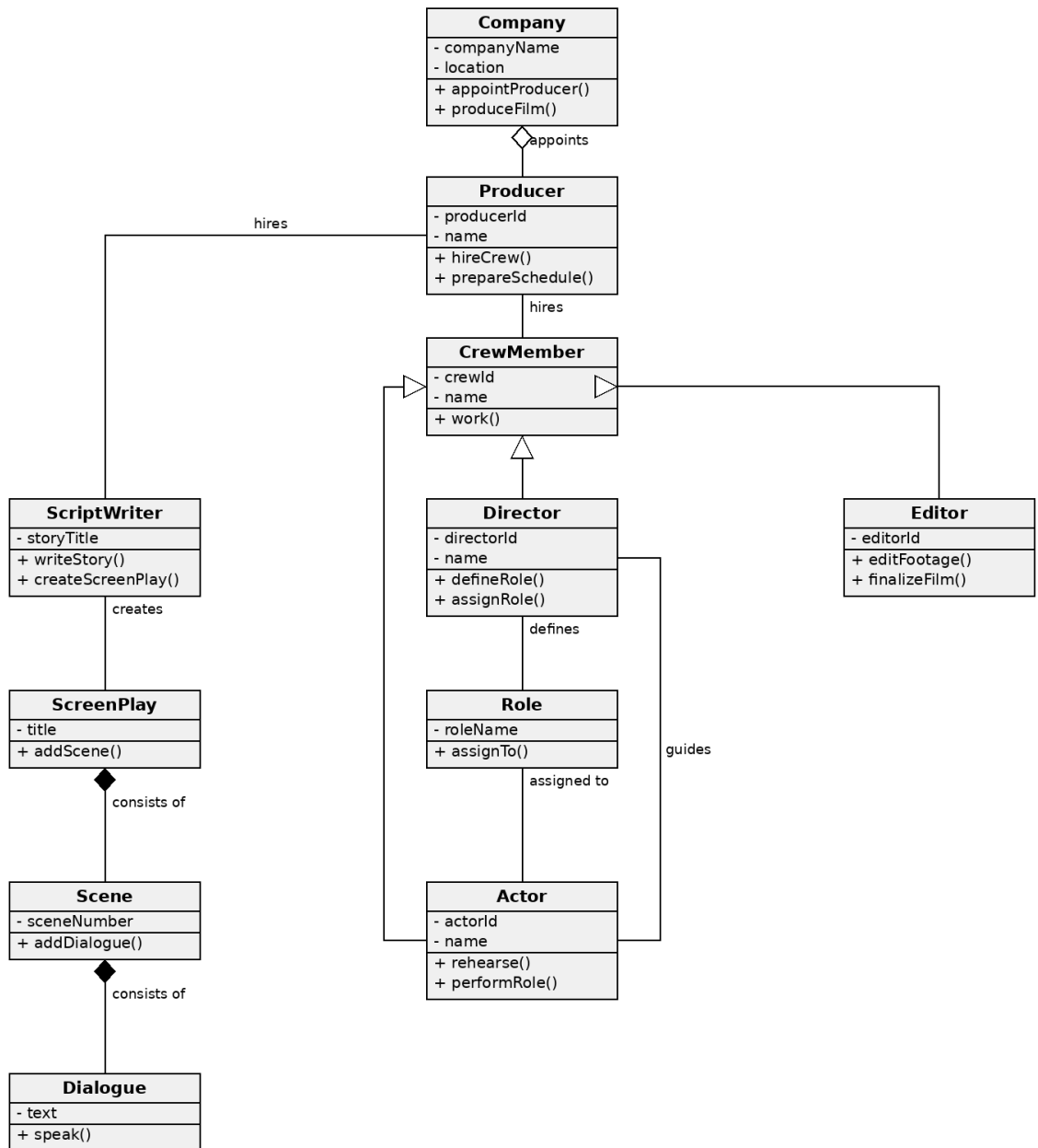
Aggregation: Company to Producer, shown with a hollow diamond. The producer is part of the production but can exist independently.

Composition: ScreenPlay to Scene and Scene to Dialogue, shown with filled diamonds. The parts cannot exist without the whole.

Note on ScriptWriter: the scenario says the producer hires a scriptwriter and other crew members, and then names the crew members as the director, the editor and the actors. ScriptWriter is therefore modeled as its own class, hired by the producer, and not as a subclass of CrewMember.

Step 5: UML Class Diagram

The complete class diagram for the Showbiz Film Production System:



Note: VU accepts assignments only as .doc or .docx files. Save your own submission file with your VU ID before uploading it to VULMS.