King Saud University

College of Computer and Information Sciences

Computer Science Department

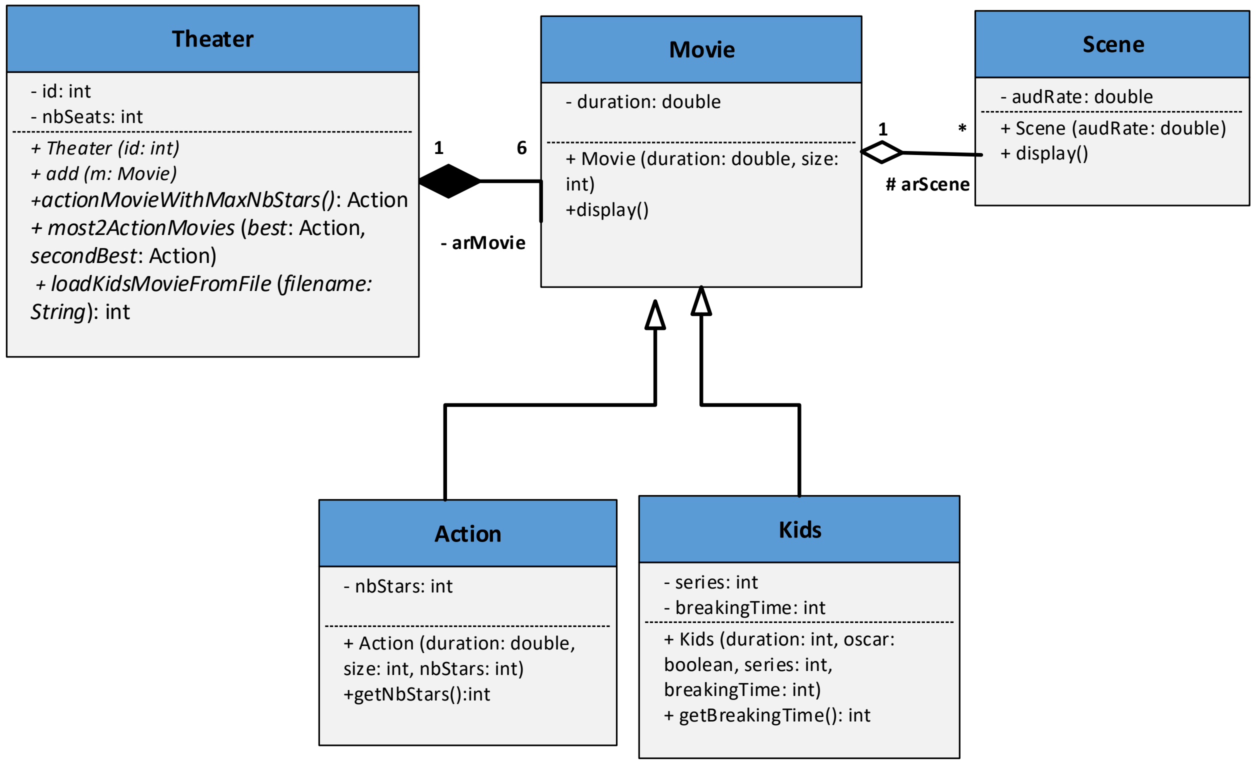
Course Code:	CSC 113
Course Title:	Computer Programming II
Semester:	Spring 2020
Exercises Cover Sheet:	**Final Exam**

Student Name:	
Student ID:	
Student Section No.	

Tick the Relevant	Computer Science B.Sc. Program ABET Student Outcomes	Question No. Relevant Is Hyperlinked	Covering %
X	**a) Apply knowledge of computing and mathematics appropriate to the computer science;**		
	b) Analyze a problem, and identify and define the computing requirements appropriate to its solution		
X	**c) Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;**		
X	**d) Function effectively on teams to accomplish a common goal;**		
	e) Understanding of professional, ethical, legal, security, and social issues and responsibilities;		
	f) Communicate effectively with a range of audiences;		
	g) Analyze the local and global impact of computing on individuals, organizations and society;		
	h) Recognition of the need for, and an ability to engage in, continuing professional development;		
X	**i) Use current techniques, skills, and tools necessary for computing practices.**		
	j) Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;		
	k) Apply design and development principles in the construction of software systems of varying complexity;		

Exercise 1:

Let's consider the following UML class diagram:


Theater
- id: int
- nbSeats: int
+ Theater (id: int)
+ add (m: Movie)
+actionMovieWithMaxNbStars(): Action
+ most2ActionMovies (best: Action, secondBest: Action)
+ loadKidsMovieFromFile (filename: String): int
1
6
- arMovie
Movie
- duration: double
+ Movie (duration: double, size: int)
+display()
1
*
# arScene
Scene
- audRate: double
+ Scene (audRate: double)
+ display()
Action
- nbStars: int
+ Action (duration: double, size: int, nbStars: int)
+getNbStars():int
Kids
- series: int
- breakingTime: int
+ Kids (duration: int, oscar: boolean, series: int, breakingTime: int)
+ getBreakingTime(): int

The class ***Scene:***

- Attributes: **audRate**: The audience rate of the scene.
- Methods:
  - ***Scene (audRate:*** *double)*: Constructor
  - ***display():*** displays all attributes of the Scene.

The class ***Movie***:

- Attributes: **duration**: The duration time of the Movie.
- Methods:
  - ***Movie(duration:*** *double*, ***size****: int)*: Constructor
  - ***display():*** displays all attributes of the Movie.

The class ***Action***:

- Attributes: **nbStars**: The number of stars (actors) of the Action movie.
- Methods:
  - **Action (** ***duration****: double*, ***size****: int*, ***nbStars****: int)*: Constructor.
  - ***getNbStars ( ):*** returns the number of stars (actors) of the Action movie.

QUESTION:

1. Implement the constructors of the class ***Movie***,
2. Implement the constructors of the class ***Action.***

Exercise 2:

Let's consider the same UML class diagram described above in Exercise 1. Let's consider that the class **Movie** is an abstract class.

The class Theater:

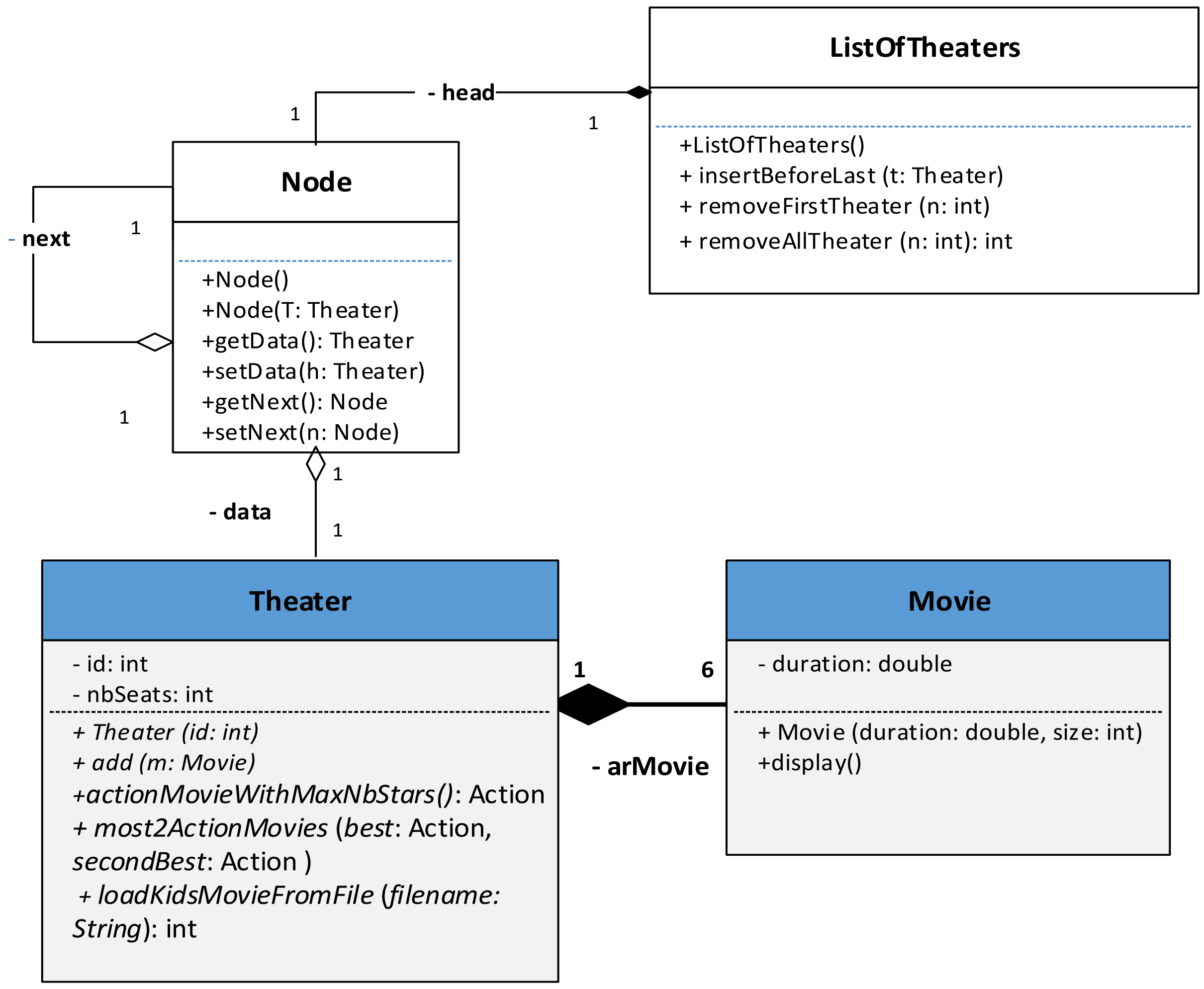
- Attributes:
  - ***id:*** The identifier of the Theater.
  - ***nbSeats:*** The number of seats of the Theater.
- Methods:
  - ***Theater (ID:*** *int)*: constructor.
  - ***add(m: Movie):*** This method adds the Movie ***m*** to the Theater. It returns an exception of type ***ArrayIndexOutOfBoundsException*** if the array of Movies is full.
  - ***actionMovieWithMaxNbStars():*** It return the Action movie that involves the maximum number of stars (actors).
  - ***most2ActionMovies*** **(*best*:** Action, ***secondBest***: Action **):** This method receives two parameters ***best*** and ***secondBest*** both of type Action movies. This method:
    - searches (i) the Action movie that has the maximum number of stars and (ii) the Action movie that has the second maximum number of stars;
    - and stores them in the parameters ***best*** and ***secondBest*** respectively.
  - ***loadKidsMovieFromFile***(***filename:*** *String*): This method reads Kids movies from a file of Movie objects named ***filename*** and adds them to the Theater. It returns the number of Kids movies added to the Theater.
  - ***getNbSeats ( )*:** returns the number of seats of the Theater.

<u>QUESTION</u>: Implement using Java the following methods of the class ***Theater:***

1. ***add***(***m:*** Movie)
2. ***actionMovieWithMaxNbStars***(): ***Action***
3. ***most2ActionMovies***(***best***: Action, ***secondBest***: Action)
4. ***loadKidsMovieFromFile***(***filename***: String): ***int***

Exercise 3:

Let’s consider the same class ***Theatre*** as described in Exercise 2.


ListOfTheaters
+ListOfTheaters()
+ insertBeforeLast (t: Theater)
+ removeFirstTheater (n: int)
+ removeAllTheater (n: int): int
- head
1
1
Node
+Node()
+Node(T: Theater)
+getData(): Theater
+setData(h: Theater)
+getNext(): Node
+setNext(n: Node)
- next
1
1
- data
1
1
Theater
- id: int
- nbSeats: int
+ Theater (id: int)
+ add (m: Movie)
+actionMovieWithMaxNbStars(): Action
+ most2ActionMovies (best: Action, secondBest: Action)
+ loadKidsMovieFromFile (filename: String): int
1
6
- arMovie
Movie
- duration: double
+ Movie (duration: double, size: int)
+display()

The class ListOfTheaters:

- Methods:
  - ***ListOfTheaters ()***: constructor.
  - ***insertBeforeLast (t***: Theater): This method will add a new theater before the last element of the linked list.
  - ***removeFirstTheater (n***: int): This method will remove the first theater in the list having a number of seats less than ***n***.
  - ***removeAllTheater (n***: int): This method will remove all theater having a number of seats less than ***n***. It returns the number of deleted theaters.

<u>**QUESTION**</u>: Implement using Java the following methods of the class ***ListOfTheaters:***

1. ***insertBeforeLast (t:*** *Theater)*
2. ***removeFirstTheater (n:*** *int)*
3. ***removeAllTheater (n:*** *int****): int***