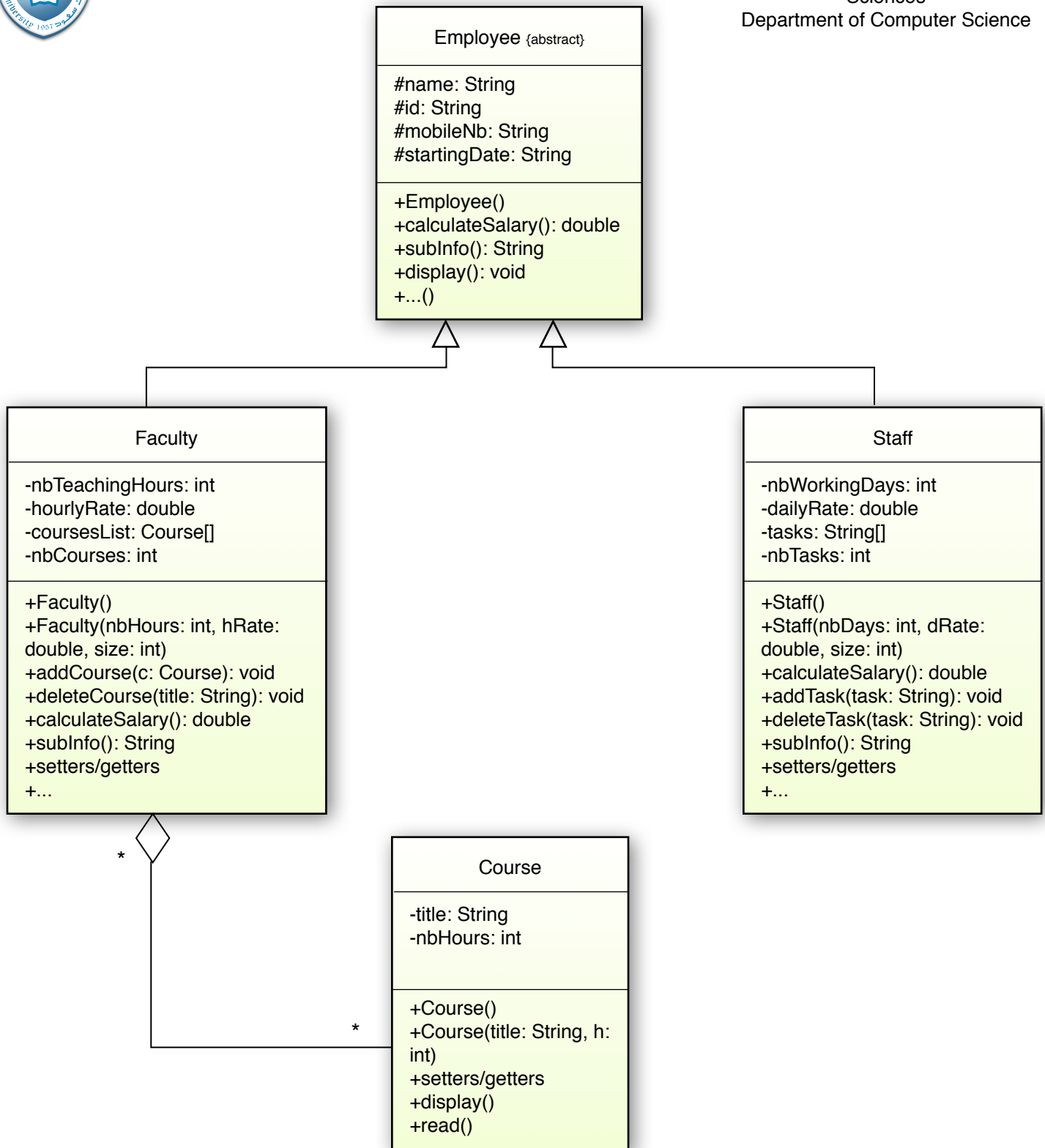


CSC 113  
Lab #6 (Inheritance & Abstract Classes)



King Saud University  
College of Computer and Information  
Sciences  
Department of Computer Science



CSC 113  
Lab #6 (Inheritance & Abstract Classes)

Class Employee (abstract)	
Attributes	Methods
<p># <i>name</i> (<i>String</i>): to hold the employee name.                      # <i>id</i> (<i>String</i>)                      # <i>mobileNb</i> (<i>String</i>)                      # <i>startingDate</i> (<i>String</i>): the first day on which the employee has started working.</p>	<p>+ <i>Employee()</i> an empty constructor.                      + <i>calculateSalary()</i> <i>double</i>: calculates and returns the salary.                      + <i>subInfo()</i> <i>String</i>: returns information of the class that implements it.                      + <i>display()</i> <i>void</i>: it should display all the possible information, and it should use <i>subInfo()</i> in order to get the sub-classes information.</p>
Class Staff	
Attributes	Methods
<p>- <i>nbWorkingDays</i> (<i>int</i>): to hold the number of days that a staff should work.                      - <i>dailyRate</i> (<i>double</i>): to hold the amount of money per day that should be paid to a staff.                      - <i>tasks</i> [] (<i>String</i>): an array contains a list of a staff's tasks.                      - <i>nbTasks</i> (<i>int</i>): the current number of tasks.</p>	<p>+ <i>Staff()</i> an empty constructor.                      + <i>Staff(nbDays int, dRate double, size, int)</i> a copy constructor.                      + <i>calculateSalary()</i> <i>double</i>: it should calculate a staff's salary, which is: the number of working days x daily rate.                      + <i>addTask (task String) void</i>: it should print the addition result (whether it's added or not).                      + <i>deleteTask (task String) void</i>: it should print the deletion result (whether it's deleted or not).                      + <i>subInfo()</i> <i>String</i>: should return a string that contains all the staff information (attributes).                      + setters/getters.</p>
Class Faculty	
Attributes	Methods
<p>- <i>nbTeachingHours</i> (<i>int</i>): to hold the number of hours that a faculty should teach.                      - <i>hourlyRate</i> (<i>double</i>): to hold the amount of money per hour that should be paid to a faculty.                      - <i>coursesList</i> [] (<i>Course</i>): an array contains a list of faculty's teaching courses.                      - <i>nbCourses</i> (<i>int</i>): the current number of courses.</p>	<p>+ <i>Faculty()</i> an empty constructor.                      + <i>Faculty(nbHours int, hRate double, size, int)</i> a copy constructor.                      + <i>addCourse (c Course) void</i>: it should print the addition result (whether it's added or not).                      + <i>deleteTask (title String) void</i>: it should print the deletion result (whether it's deleted or not).                      + <i>calculateSalary()</i> <i>double</i>: it should calculate a faculty's salary, which is: number of teaching hours x hourly rate.                      + <i>subInfo()</i> <i>String</i>: should return a string that contains all the faculty information (attributes).                      + setters/getters.</p>

Implement all these four classes in java, and also implement a class that contains a main method to test your code. In the main method, you can create several *Course* objects, and assign them to several pre-created *Faculty* objects, in addition to creating some *Staff* objects and assigning them some tasks. After that you can call each object's display method.