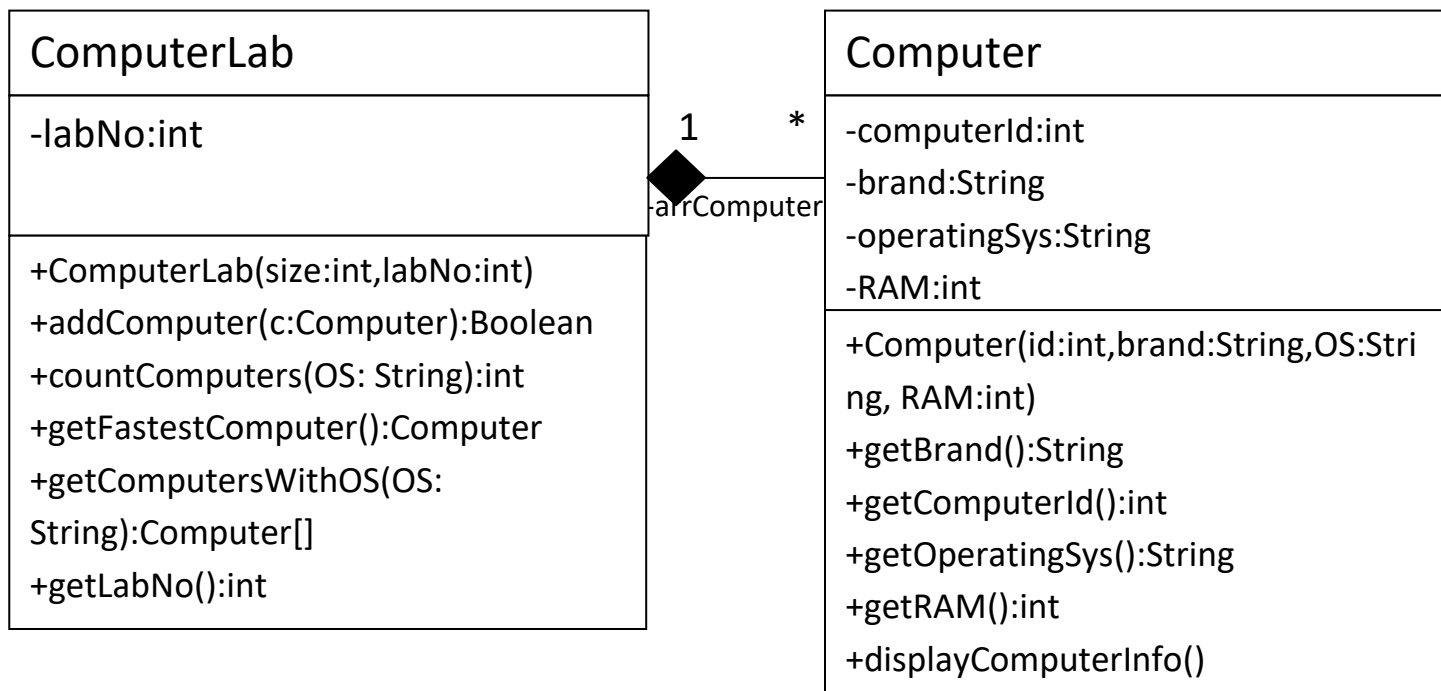




Lab: **Relationship between Classes (Composition)**

Create the classes along with the functionality given in the following UML Diagram. To understand the problem, please refer to the description given after the diagram.



**Computer Class:**

o Attributes:

- computerId: unique id of the computer.
- **brand**: the brand of the computer e.g. Sony, HP, LG etc.
- **operatingSys**: The operating system of the computer. e.g. Windows, Linux, Mac etc.
- **RAM**: RAM in Giga Bytes that the computer has.

o Methods:

- **Computer(id:int, brand: string, operatingSys: String, RAM: int)**: constructor
- **displayComputerInfo()**: this method displays all the attributes of the computer.
- **Getters**



**King Saud University**  
**College of Computer and Information Systems**  
**Department of Computer Science**  
**CSC 113: Java Programming-II**



**Lab: Relationship between Classes (Composition)**

***ComputerLab Class:***

○ Attributes:

- ***labNo***: Unique Number of the ComputerLab.

○ Methods:

- ***ComputerLab(size:int, labNo:int)***: constructor
- ***addComputer(c:Computer)***: this method adds a computer to the ComputerLab. It returns true if the *Computer c* is added; false otherwise.
- ***countComputer(OS:String)***: this method returns the number of computers of the Computer Lab with operating system *OS*.
- ***getFastestComputer()***: this method returns the computer who has the maximum RAM in the Computer Lab.
- ***getComputersWithOS(OS: String)***: this method returns an array containing all computers of the Computer Lab that have the Operating System *OS*.
- ***getLabNo()***: returns the unique lab number.

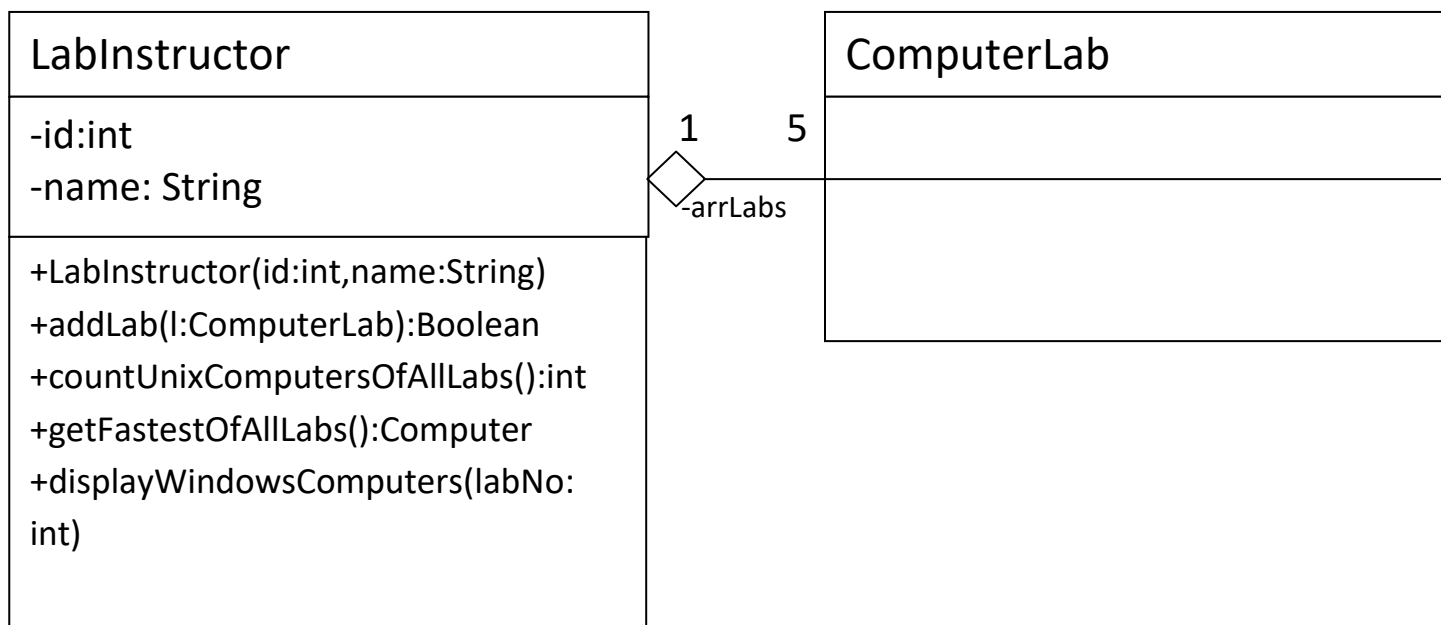
**Ex1: Write the code for Computer and ComputerLab classes.**

**(See Next Pages for Ex2 and Ex3)**



Lab: Relationship between Classes (Composition)

Ex2: Consider the class `ComputerLab` created in Exercise 1 and write the code for class `LabInstructor`



**LabInstructor Class:**

○ Attributes:

- **labNo:** Unique Number of the ComputerLab.

○ Methods:

- **LabInstructor(id:int,name:String):** constructor
- **addLab(l:ComputerLab):** this method assigns a computer lab to Instructor. It returns true if the *ComputerLab l* is assigned; false otherwise.
- **countUnixComputersOfAllLabs():** this method returns the number of unix computers in all labs.
- **getFastestofAllLabs():** this method returns the computer who has the maximum RAM in all Computer Labs assigned to the instructor.



**King Saud University**  
**College of Computer and Information Systems**  
**Department of Computer Science**  
**CSC 113: Java Programming-II**



**Lab: Relationship between Classes (Composition)**

- *displayWindowsComputers(LabNo: int)*: this method displays the attributes of the windows computers in Lab No *LabNo*.

**Ex3: Use the following code to create the Main class to test already created classes. Also compare your output with the output given after the code and try to find your mistakes.**

Code:

```
public class Main {  
  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        Computer c1=new Computer(1,"Sony","Windows",4);  
        Computer c2=new Computer(2,"Sony","Windows",6);  
        Computer c3=new Computer(3,"Sony","Windows",8);  
        Computer c4=new Computer(4,"HP","Unix",4);  
        Computer c5=new Computer(5,"Sony","Unix",12);  
  
        Computer c6=new Computer(6,"Sony","Windows",10);  
        Computer c7=new Computer(7,"HP","Unix",4);  
        Computer c8=new Computer(8,"HP","Unix",16);  
  
        ComputerLab L1=new ComputerLab(26,2);  
        System.out.println(L1.addComputer(c1));  
        System.out.println(L1.addComputer(c2));  
        System.out.println(L1.addComputer(c3));  
  
        ComputerLab L2=new ComputerLab(40,3);  
        System.out.println(L2.addComputer(c3));  
        System.out.println(L2.addComputer(c4));  
        System.out.println(L2.addComputer(c5));  
  
        ComputerLab L3=new ComputerLab(112,3);  
        System.out.println(L3.addComputer(c6));  
        System.out.println(L3.addComputer(c7));  
        System.out.println(L3.addComputer(c8));  
    }  
}
```



**King Saud University**  
**College of Computer and Information Systems**  
**Department of Computer Science**  
**CSC 113: Java Programming-II**



**Lab: Relationship between Classes (Composition)**

```
LabInstructor inst=new LabInstructor(1001,"Mudassar");

System.out.println(inst.addLab(L1));
System.out.println(inst.addLab(L2));
System.out.println(inst.addLab(L3));

System.out.println("\n\nFollowing are Windows Computers in Lab
26: ");
inst.displayWindowsComputers(26); // two windows computers

System.out.println("\n\nTotal Number of Unix Computers in All
Labs :"+ inst.countUnixComputersOfAllLabs()); // 4 Unix Computers

Computer fastest=inst.getFastestOfAllLabs();
System.out.println("\n\nBased on RAM, Fastest Computer in all
labs is following : ");
fastest.displayComputerInfo();

}

}
```

Output:

```
true
true
false
true
true
true
true
true
true
true
true
true
true
```

```
Following are Windows Computers in Lab 26:
Computer Id: 1
Brand: Sony
Operating System: Windows
RAM in GBs: 4
Computer Id: 2
Brand: Sony
Operating System: Windows
RAM in GBs: 6
```



**King Saud University**  
**College of Computer and Information Systems**  
**Department of Computer Science**  
**CSC 113: Java Programming-II**



**Lab: Relationship between Classes (Composition)**

Total Number of Unix Computers in All Labs :4

Based on RAM, Fastest Computer in all labs is following :

Computer Id: 8

Brand: HP

Operating System: Unix

RAM in GBs: 16