Batch: A3   Roll No.: 16010121045
Experiment / assignment / tutorial No.06
Grade: AA / AB / BB / BC / CC / CD /DD
Signature of the Staff In-charge with date

TITLE :Case Study (for Class Diagram)

AIM: Draw class Diagram for the chosen Case Study . Clearly show
●   Attributes
●   Multiplicities between classes
●   Aggregations/compositions/Association between classes
●   Generalization between classes in the class diagram.
And show the implementation of aggregation, association, composition and generalization between the classes.

Expected OUTCOME of Experiment:
CO1: Understand the features of object oriented programming compared with

procedural approach with C++ and Java.

CO2: Explore arrays, vectors, classes and objects in C++ and Java.

CO3: Implement scenarios using object oriented concepts (Drawing class diagram, relationship between classes, sequence diagram)
CO4: Explore the interface, exceptions, multithreading, packages

Books/ Journals/ Websites referred:

1.Ralph Bravaco , Shai Simoson , "Java Programing From the Group Up" Tata McGraw-Hill.

2.Grady Booch, Object Oriented Analysis and Design .

Pre Lab/ Prior Concepts:

Define Class, Methods, Object.
Understanding of Aggregation, Association, Composition and Generalization between classes

Department of Computer Engineering

List Of Classes:
Staff: Contains the staff details

Doctor: Contains the details of doctors

Patient: Contains the patients details

Medical: Contains medicine details

Facility: Contains details about facilities available

Identify Attributes for each class:
Staff: sid, sname, desg, gender

Doctor: did, dname, specilist, doc_qual, droom

Patient: pid, pname, disease, gender, admit_status, age

Medical: med_name, exp_date, med_cost, count

Facility: fac_name

Lab: facility, lab_cost

Identify List of Methods in each classes:
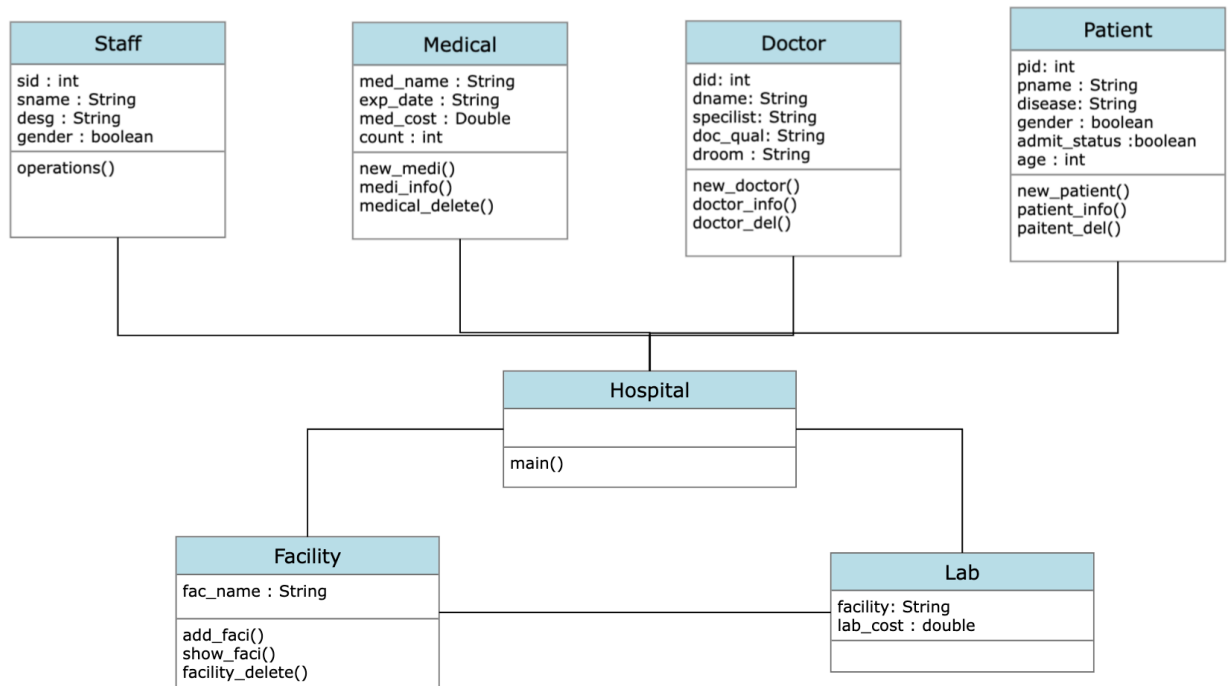Staff: new_staff(id,name,deg,gender,salary), staff_info(id), staff_del()

Doctor: new_doctor(did, dname, specilist, doc_qual, droom), doctor_info(), doctor_del()

Patient: new_patient(), patient_info(id), patient_del()

Medical: new_medi(), medi_info(id),medical_delete()

Facility: add_faci(), show_faci(id),facility_delete()

Class Diagram:

Algorithm:

1. Start
2. import java.util.* and java.util.calendar
3. Declare a class Staff
4. Declare attributes of class Staff sid, sname, desg, sex, salary.
5. Declare a method new_staff in Staff class to set the attributes for new members of the class.
6. Declare a method staff_info to print these details.
7. Declare a class Doctor
8. Declare attributes of class Doctor did, dname, specilist, appoint, doc_qual, droom
9. Declare method new_doctor and doctor_jnfo which has same purpose as class Staff methods had for Staff.
10. In the same way as done in previous steps declare classes Patient, Medical, Facility, Lab
11. Declare following attributes for the respective classes

    Patient: pid, pname, disease, sex, admit_status, age
    Medical: med_name, med_comp, exp_date, med_cost, count Facility: fac_name
    Lab: facility, lab_cost.

12. Now, declare following methods for the classes and these methods serve the same purpose as described for class Staff and Doctor i.e. to set the attributes for new members of the class and to print the details.

    Patient: new_patient(), patient_info() Medical: new_medi()
    Facility: add_faci(), show_faci() Lab: new_faci(), faci_list()

Department of Computer Engineering

13. In the main method define a String containing months names
14. Create a object calendar for calendar.
15. Declare variables count1, count2, count3, count4, count5,count6.
16. Print current date and time.
17. Create Array of objects for the respective classes.
18. Initialize the for loops for the classes which are used to add new objects to the class.
19. Add some objects to this array list.
20. Get choice from the user.
21. If 1 is selected then user gets two options 1. Add New Entry 2. Existing Doctorslist
22. After to go back enter 1 and 0 for main menu.
23. According to choose entered by the user add the objects to the class or display the details using the methods defined earlier.
24. Repeat step 22 and 23 for all other options.
25. If user enters 1.
26. Exit

Conclusion : Successfully completed the given task.

Date: _____   **Signature of faculty in-charge**

Post Lab Descriptive Questions

1. Consider the following class:

```
public class TypeOfVariable{
public static int a;
int b,c;
public void printValue(){
        int x = 10;
}
public static void main(String args[]){
        TypeOfVariable object=new TypeOfVariable();
        object.printValue();
}
}
```

a). What are the class/static variables?

A static variable is common to all the instances (or objects) of the class because it is a class level variable. In other words you can say that only a single copy of static variable is created and shared among all the instances of the class. Memory allocation for such variables only happens once when the class is loaded in the memory. Static variables are also known as Class Variables.
Unlike non-static variables, such variables can be accessed directly in static and non-static methods.

Here **a** is static variable.

b). What are the instance variables?

Instance variable in Java is used by Objects to store their states. Variables which are defined without the STATIC keyword and are Outside any method declaration are Object-specific and are known as instance variables. They are called so because their values are instance specific and are not shared among instances.

Here **b** and **c** are instance variables.

c.)What are local variables?

A local variable in Java is a variable that's declared within the body of a method. Then you can use the variable only within that method. Other methods in the class aren't even aware that the variable exists. You don't specify static on a declaration for a local variable.

Here **x** is a local variable.

2.What is the output from the following code:

```java
public class Test
{
    static int x = 11;
    private int y = 33;
    public void method1(int x)
    {
        Test t = new Test();
        this.x = 22;
        y = 44;

        System.out.println("Test.x: " + Test.x);
        System.out.println("t.x: " + t.x);
        System.out.println("t.y: " + t.y);
        System.out.println("y: " + y);
    }

    public static void main(String args[])
    {
        Test t = new Test();
        t.method1(5);
    }
}
```

Output

Test.x = 22 t.x = 22
t.y = 33
y = 44