**Batch: B3 Roll No.: 121**

**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:**

DefineClass, Methods, Object.

Understanding of Aggregation, Association, Composition and Generalization between classes

**List Of Classes:**

1. Member
2. Product
3. detail\_order
4. Order\_Transaction
5. Payment

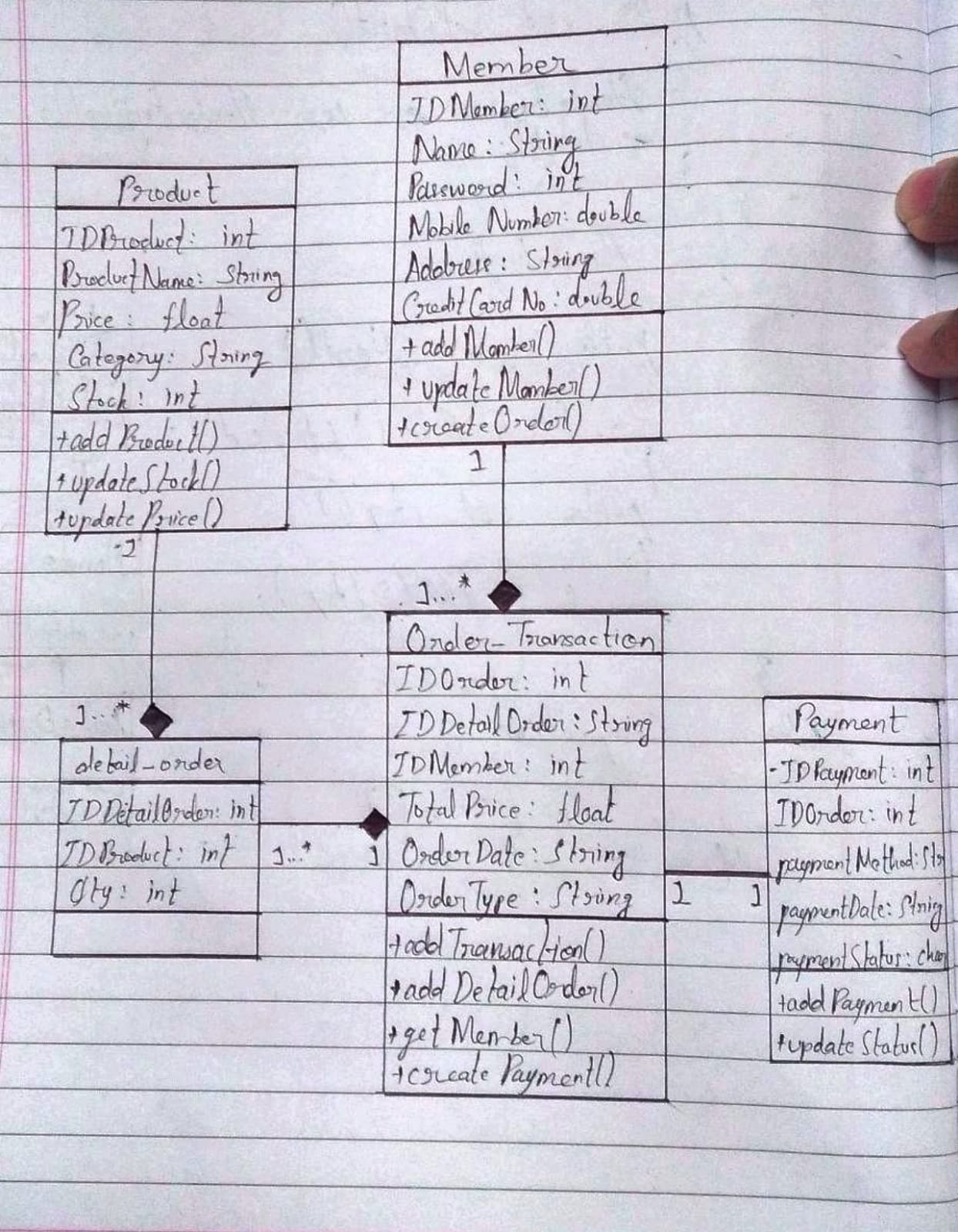
**Identify Attributes for each class:**

1. For Member class, the attributes are:
2. IDMember of type integer.
3. Name of type String.
4. Password of type integer.
5. Mobile Number of type double.
6. Address of type String.
7. CreditCardNo of type double.
8. For Product class, the attributes are:
9. IDProduct of type integer.
10. ProductName of type String.
11. Price of type float.
12. Category of type String.
13. Stock of type integer.
14. For detail\_order class, the attributes are:
15. IDDetailOrder of type integer.
16. IDProduct of type integer.
17. Qty of type integer.
18. For Order\_Transaction class, the attributes are:
19. IDOrder of type int.
20. IDDetailOrder of type String.
21. IDMember of type integer.
22. TotalPrice of type float.
23. OrderDate of type String.
24. OrderType of type String.
25. For Payment class, the attributes are:
26. IDPayment of type integer.
27. IDOrder of type integer.
28. paymentMethod of type String.
29. paymentDate of type String.
30. paymentStatus of type character.

**Identify List of Methods in each classes:**

1. For Member class, the list of methods is:
2. addMember(), which is public.
3. updateMember(), which is public.
4. createOrder(), which is public.
5. For Product class, the list of methods is:
6. addProduct(), which is public.
7. updateStock(), which is public.
8. updatePrice(), which is public.
9. For detail\_order class, the list of methods is:
10. No methods have been defined in this class.
11. For Order\_Transaction class, the list of methods is:
12. addTransaction(), which is public.
13. addDetailOrder(), which is public.
14. getMember(), which is public.
15. createPayment(), which is public.
16. For Payment class, the list of methods is:
17. addPayment(), which is public.
18. updateStatus(), which is public.

**Class Diagram:**



**Conclusion**

Thus, in this experiment, the concept of Class Diagram was learnt. Class Diagrams enable a programmer to visualize the program requirements. Further, internationally standard rules are followed while making a class diagram so that the idea of one programmer can be easily communicated to another programmer working on the same code. Thus, there is better co-ordination among team members which makes possible the creation and maintenance of huge code for big industrial purposes like factories, banking, etc.

**Date: \_\_22-11-22\_\_ 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?

Ans. The class/static variable is **a**.

b). What are the instance variables?

Ans. The instance variables are **b** and **c**.

c.)What are local variables?

Ans. The local variable is **x**.

**2.What is the output from the following code:**

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);

    }

}

Ans.

Test.x: 22

t.x: 22

t.y: 33

y: 44