Name → ABHISHEK DHAR
ROLL → 23201
BATCH → E-10

ASSIGNMENT NO - 03

Title → Inheritance

Aim → Design and develop inheritance for a given case study, identify objects and relationships and implement inheritance whenever applicable. Employee class has Emp_name, Emp_id, Address, mail_id and mobile_no as members. Inherit classes Programmer, Team Lead, Assistant project manager and project manager from employee class. Add Basic pay (BP) as the member of all the inherited classes with 97% of BP as DA. 10% of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.

Objectives → To learn the concepts of hierarchial inheritance.

Theory → Hierarchial inheritance

A) **Definition** →

When a class has more than one child classes or in other words more than one child classes have same parent class then this type of inheritance is known as Hierarchial inheritance.

B) Explanation →

① To use inheritance in java 'extends' keyword is used 'extends' is used to inherit properties of base class to child class.

② Syntax →

```
class Super { }
class sub extends super { }
```

Hierarchial inheritance →

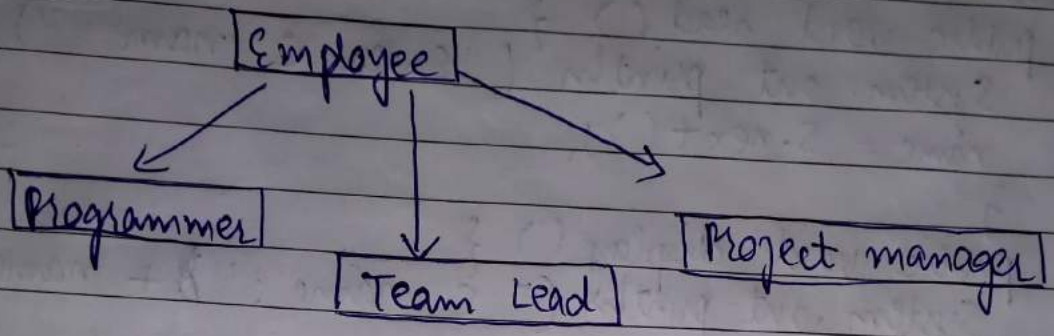
③ In hierarchial inheritance, one class serves as a superclass for more than one sub class.

④ Inheritance is when an object or class is based on another object or class, using same implementation specifying implementation to maintain same behavior.

⑤ It is a mechanism for code reuse & allow independent extensions of the orginal software via public classes and interfaces.

⑥ The relationships of objects and classes through inheritance give rise to a hierarchy.

C) Diagram →



D) Example →

```
class A {
 String name;
 int Emp-id;
 public void read () {
  System.out.println ("Enter name:");
  name = S.next();
 }
 public void display () {
   System.out.println ("Name: " + name);
 }
}

class B extends A {
 public void read () {
   System.out.println ("Enter B name:");
   name = S.next();
 }
 public void display () {
   System.out.println ("B name:" + name);

}
```

```
class C extends A {
  public void read () {
    System.out.println ("Enter C name :");
    name = S.next();
  }
  public void display () {
    System.out.println (" C name :" + name);
  }
}

class main_method {
  public static void main (String [] args) {
    B obj1 = new B();
    C obj2 = new C();
    obj1.read();
    obj1.display();
    obj2.read();
    obj2.display();
  }
}
```

O/P →

```
Enter B name : Janhavi
B name : Janhavi
Enter C name : Varada
C name : Varada
```

E) Working →

The four classes Programmer, Team Lead, Assistant Project manager and Project manager are the Subclasses inheriting from Employee class. Hence, all the methods and datamembers of Employee are accessible in sub classes. So in the above example name is used in all classes to read and display its values. In main method, according to object declared, coresponding read and display function would be called and executed.

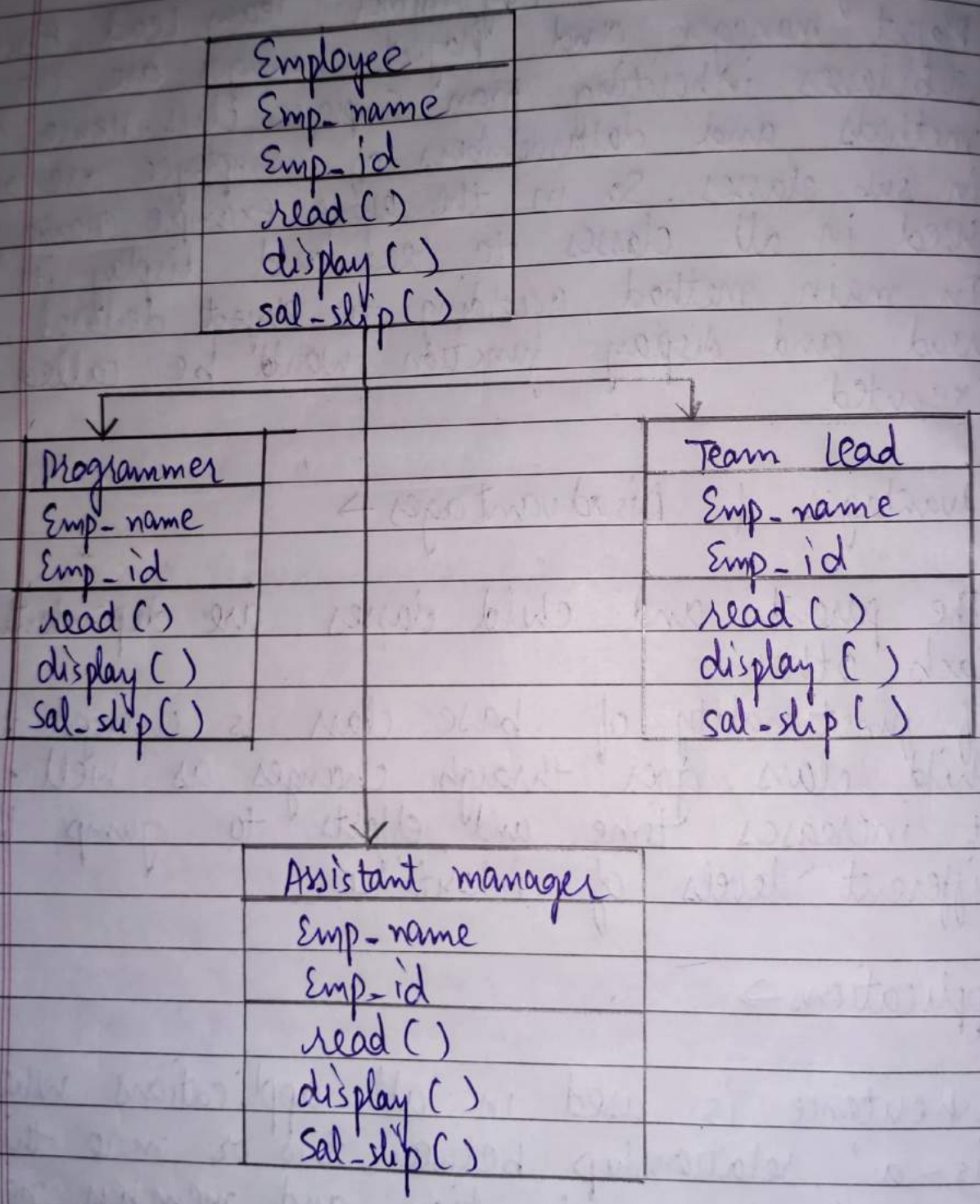
F) Advantages & Disadvantages →

a) The parent and child classes are dependent on each other.
b) If functionality of base class is changed then child class goes through changes as well.
c) It increases time and efforts to jump through different levels of inheritance.

G) Applications →

(1) Inheritance is used in all applications where there 'is-a' relationship between two or more things
(2) Also used whenever function, and members need to be used in various diff. setting.

Class Diagram →



Conclusion →

Learnt to identify relation among objects and apply concept of hierarchial inheritance.