ASSIGNMENT

B. LIKHITHA

Assignment 1: Person → Student

Create:

• Person class → name, age

• Student class → rollNo

• Display all details using child object

Concepts: Basic inheritance

Person.java | Share | Run | Output

```
class Person {
    String name;
    int age;

    public static void main(String[] args) {
        Student s = new Student();
        s.name = "Likhitha";
        s.age = 20;
        s.rollNo = 101;
        s.display();
    }
}
class Student extends Person {
    int rollNo;
    void display() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Roll No: " + rollNo);
    }}
```

```
Name: Likhitha
Age: 20
Roll No: 101

=== Code Execution Successful ===
```

Assignment 2: Vehicle → Car

Create:

• Vehicle → speed

• Car → brand

• Use inherited variable in child

Concepts: Property inheritance

Vehicle.java

Share Run Output

```java
class Vehicle {
    int speed;
    public static void main(String[] args) {
        Car c = new Car();
        c.brand = "BMW";
        c.speed = 220;

        c.showDetails();
    }
}

class Car extends Vehicle {
    String brand;

    void showDetails() {
        System.out.println("Brand: " + brand);
        System.out.println("Speed: " + speed);
    }
}
```

```
Brand: BMW
Speed: 220

=== Code Execution Successful ===
```

Assignment 3: Animal → Dog

Create:

• Animal → eat()

• Dog → bark()

• Call both methods using Dog object

Concepts: Method inheritance

Animal.java

Share Run Output

```java
class Animal {
    void eat() {
        System.out.println("Animal is eating...");
    }

    public static void main(String[] args) {
        Dog d = new Dog();
        d.eat();
        d.bark();
    }
}

class Dog extends Animal {
    void bark() {
        System.out.println("Dog is barking...");
    }
}
```

```
Animal is eating...
Dog is barking...

=== Code Execution Successful ===
```

Assignment 4: Employee → Manager

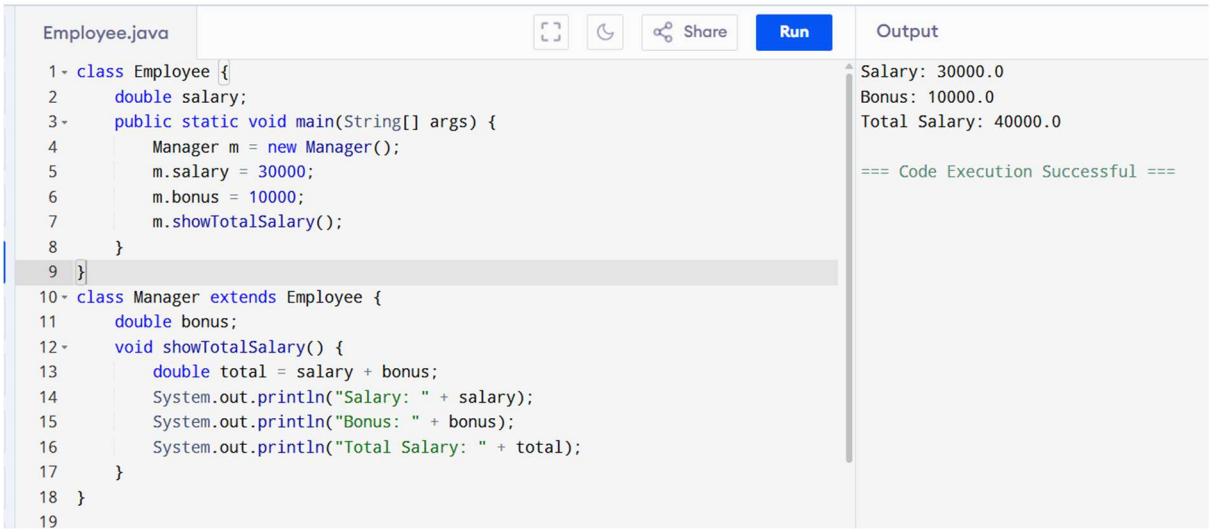
Create:

• Employee → salary

• Manager → bonus

• Calculate total salary

Concepts: Code reuse



Assignment 5: Bank → SavingAccount

Create:

• Bank → interestRate

• SavingAccount → calculateInterest()

Concepts: Parent data usage

Assignment 6: Constructor Inheritance

Create:

• Person constructor initializes name

• Student constructor initializes rollNo

• Use super() to call parent constructor

Concepts: super() keyword

Person.java

Share Run

```java
class Person {
    String name;
    Person(String name) {
        this.name = name; }
    public static void main(String[] args) {
        Student s = new Student("Likhitha", 101);
        s.show(); }}
class Student extends Person {
    int rollNo;
    Student(String name, int rollNo) {
        super(name);
        this.rollNo = rollNo;
    }
    void show() {
        System.out.println("Name: " + name);
        System.out.println("Roll No: " + rollNo);
    }
}
```

Output

```
Name: Likhitha
Roll No: 101

=== Code Execution Successful ===
```