Name	Shubhan Singh
UID no.	2022300118
Experiment No.	8-A

PROBLEM STATEMENT :	Ankit works at ABC Company. He noticed that different roles(positions) have different salaries and bonuses. The 1st Role is an 'Intern' which has 3/4th of the base salary of an Employee. Then there is 'Clerk' which has ½ of base salary. And then there are 'Manager' who have twice the base salary of that of an employee. Help him write a program in Java as follows. Create a class 'Employee' which has a method named 'getSalary' which returns a base salary of Rs. 10,000. It also has methods named 'getBonus' which returns the bonus amount for that role(initially set to Rs. 0). Make 3 subclasses for different roles which inherit the 'Employee' class and each has functions named 'getSalary' and 'getBonus'.(You can assume values for 'getBonus' method) Display the output for all cases. Also print the total salary received by each Employee after getting the bonus. Note : Solve using method overriding
THEORY:	In object-oriented programming (OOP), "override" refers to the ability of a subclass to provide a different implementation of a method that is already defined in its superclass. This allows the subclass to customize or specialize the behavior of the inherited method to suit its specific needs. The overridden method in the subclass replaces the implementation inherited from the superclass. When a method is overridden, the version of the method to be executed is determined by the actual object type at runtime. This is known as dynamic method dispatch or late binding. It means that when a method is called on an object, the programming language determines which version of the method to invoke based on the actual type of the object, rather than the type

	declared at compile-time. Here are some important points to understand about method overriding: Inheritance: Method overriding is possible because of the concept of inheritance in object-oriented programming. Inheritance allows a subclass to inherit properties and behaviors (including methods) from its superclass. Signature and Access Modifier: To override a method, the subclass must provide the same method signature as the superclass. The signature includes the method name, parameter types, and the return type. Additionally, the access modifier of the overriding method in the subclass must be the same or more accessible than the method being overridden. @Override Annotation (Java): In Java, you can use the @Override annotation before the method declaration in the subclass to explicitly indicate that you are intending to override a method. This annotation helps in catching potential errors at compile-time if the method being overridden doesn't exist in the superclass. Polymorphism: Method overriding is closely related to the concept of polymorphism. Polymorphism allows objects of different classes to be treated as objects of a common superclass. By overriding methods, you can achieve different behaviors for objects of the same superclass, depending on their actual types. It's important to note that not all methods can be overridden. Whether a method is eligible for overriding depends on the language and the specific rules of the programming language you are using. Method overriding is a fundamental concept in object-oriented programming and is widely used to achieve abstraction, modularity, and code reuse. It allows for flexible and extensible designs by enabling subclasses to provide their own implementation while leveraging the common interface and behavior defined in the superclass.

PROGRAM:

```
import java.util.Scanner;

class Employee {
    float salary, bonus;

    public Employee() {
        salary = 10000;
        bonus = 0;
    }

    public void setBaseSalary(float salary) {
        this.salary = salary;
    }

    public float getSalary() {
        return salary + salary * bonus / 100;
    }

    public float getBonus() {
        return bonus;
    }

    public void setBonus(float bonus) {
        this.bonus = bonus;
    }

}

class Intern extends Employee {
    float internbonus;

    Intern() {
        internbonus = 0;
    }

    @Override
    public float getSalary() {
        return 3 * (salary + salary * internbonus / 100) / 4;
    }

    @Override
    public void setBonus(float bonus) {
        internbonus = bonus;
    }
    @Override
    public float getBonus() {
        return internbonus;
    }
}

class Clerk extends Employee {
    float Clerkbonus;

    Clerk() {
        Clerkbonus = 0;
    }
```

```
    @Override
    public float getSalary() {
        return (salary + salary * Clerkbonus / 100) / 2;
    }

    @Override
    public void setBonus(float bonus) {
        Clerkbonus = bonus;
    }
    @Override
    public float getBonus() {
        return Clerkbonus;
    }
}

class Manager extends Employee {
    float Managerbonus;

    Manager() {
        Managerbonus = 0;
    }

    @Override
    public float getSalary() {
        return 2 * (salary + salary * Managerbonus / 100);
    }

    @Override
    public void setBonus(float bonus) {
        Managerbonus = bonus;
    }
    @Override
    public float getBonus() {
        return Managerbonus;
    }
}

public class employees {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Intern intern1 = new Intern();
        Clerk clerk1 = new Clerk();
        Manager manager1 = new Manager();
        System.out.println("Enter values of bonus for Intern,
clerk and manager respectively:");
        intern1.setBonus(sc.nextFloat());
        clerk1.setBonus(sc.nextFloat());
        manager1.setBonus(sc.nextFloat());
        System.out.printf("The salary of Intern is: %.2f
(with %.2f%%
bonus)\n",intern1.getSalary(),intern1.getBonus());
        System.out.printf("The salary of Clerk is: %.2f (with
%.2f%% bonus)\n",clerk1.getSalary(),clerk1.getBonus());
        System.out.printf("The salary of Manager is: %.2f
(with %.2f%%
```

```
bonus)\n",manager1.getSalary(),manager1.getBonus());
        sc.close();
    }
}
```

RESULT:

```
Enter values of bonus for Intern, clerk and manager respectively:
5 7 3.45
The salary of Intern is: 7875.00 (with 5.00% bonus)
The salary of Clerk is: 5350.00 (with 7.00% bonus)
The salary of Manager is: 20690.00 (with 3.45% bonus)

Process finished with exit code 0
```