Write a java program which has a class named employer that has two public members, employeename and employeeid and two private members , hours and rates. The class employer has getter methods to get the private members and a method named print which will print the employeename and employeeid. Now a subclass of employer named employee will have a constructor which will assign the values to the members of its superclass and it has a method named wage which will calculate the regular wage. Create objects of employee class inside the main class and call the methods.

In inheritance, a subclass cannot directly inherit private members from its superclass. If you want to use the private members of superclass in the subclass, you have to do the following:

1. First create a superclass constructor which will assign the values to its private members
2. Use the super keyword with correct number of parameters inside the subclass constructor. Remember you have to use super as the first statement inside subclass constructor
3. Create getter methods for the private members of superclass so that you can use the private members inside the methods of a subclass.

```
class employer{
    String employeename;
    int employeeid;
    private double rate;
    private int hours;
    employer(double rate, int hours){// we create the
superclass constructor to assign values to the
private members
        this.rate=rate;
        this.hours=hours;
    }

    double getrate() {
        return rate;
    }
    int gethours() {
        return hours;
    }
    void print() {
        System.out.println("Employee name is
"+employeename);
        System.out.println("employee id is
"+employeeid);
```

```
    }

}
class employee extends employer {
    employee(String employeename, int employeeid,
double rate, int hours){
        super(rate,hours);// we used super so that it
can inherit superclass constructor to assign the
values to the private members.
        this.employeename=employeename;
        this.employeeid=employeeid;

        }
    double wage() {
        double totalwage;
        totalwage=gethours()*getrate();// use the
getter methods for the private members of superclass
        return totalwage;
    }

}

public class inheritanceprivate {

    public static void main(String[] args) {
        employee e= new employee("Rafsun", 101, 4.5,
10);// TODO Auto-generated method stub
        e.print();
        System.out.println(e.wage());
    }

}
```

Write a Java program which has a class named bank that has two public members, accountname and accountid and one private member named balance. Bank class has a method named print which will print out the accountname and accountid. A subclass of bank named checkbalance will have a constructor which will assign the values to the superclass members and it will have three methods, deposit which will deposit money to the bank account, withdraw which will withdraw money from the bank account and balance which will show the total balance after deposit and withdraw. Create object of the subclass and call the methods.

```
class bank{
    String accountname;
    int accountid;
    private double balance;
    bank(double balance){
        this.balance=balance;
    }
    double getbalance() {
        return balance;
    }
    void print() {
        System.out.println(accountname);
        System.out.println(accountid);
    }
}
class checkbalance extends bank{
    double b=getbalance();
    checkbalance(String accountname, int accountid,
double balance){
        super(balance);
        this.accountname=accountname;
        this.accountid=accountid;
    }
    double deposit(double amount) {
        return b+=amount;

    }
    double withdraw(double amount) {
        if(amount>getbalance()) {
            return 0;
        }
        else {
            return b-=amount;
        }
    }
    double currentbalance() {
        return b;
    }
}
public class inheritance_bank {

    public static void main(String[] args) {
        checkbalance c= new checkbalance("Rafsun",
101, 1500);// TODO Auto-generated method stub
        c.print();
        c.deposit(1000);
        c.withdraw(500);
        System.out.println(c.currentbalance());
    }

}
```