1. Write a program that describes the hierarchy of an organization. Here we need to write 3 classes Employee, Manager & Labour where Manager & Labour are the sub classes of the Employee. Manager has incentive &Labour has over time. Add the functionality to calculate total salary of all the employees. Use polymorphism i.e. method overriding.

**package** com.zensar;

**public** **class** Employee {

**void** managersalary(**int** hours) {

**int** employee\_salary\_per\_day=100\*hours;

System.***out***.println("manager salary for per day is :"+employee\_salary\_per\_day);

}

**void** laboursalary(**int** hours)

{

**int** labour\_salary\_per\_day=25\*hours;

System.***out***.println("labour salary for per day is :" +labour\_salary\_per\_day);

}

}

**package** com.zensar;

**public** **class** Manager **extends** Employee{

@Override

**void** managersalary(**int** hours) {

// **TODO** Auto-generated method stub

**super**.managersalary(hours);

}

}

**package** com.zensar;

**public** **class** Labour **extends** Employee{

@Override

**void** managersalary(**int** hours) {

// **TODO** Auto-generated method stub

**super**.managersalary(hours);

}

@Override

**void** laboursalary(**int** hours) {

// **TODO** Auto-generated method stub

**super**.laboursalary(hours);

}

**public** **static** **void** main(String args[])

{

Labour obj1=**new** Labour();

obj1.managersalary(9);

obj1.laboursalary(12);

}

}

Output:

manager salary for per day is :900

labour salary for per day is :300

1. Write a program to consider saving & current account in the bank. Saving account holder has ‘Fixed Deposits’ whereas Current account holder has cash credit. Apply polymorphism to find out total cash in the bank.

**package** com.zensar;

**public** **class** BankAccount {

**private** String name;

**protected** **double** balance;

**public** String getName() {

**return** **this**.name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **void** deposite(**double** amount) {

**this**.balance += amount;

}

**public** **boolean** withdraw(**double** amount) {

**if** (balance > amount) {

balance -= amount;

**return** **true**;

} **else** {

**return** **false**;

}

}

}

**package** com.zensar;

**public** **class** CurrentAccount **extends** BankAccount {

**public** **boolean** withdraw(**double** amount) {

balance -= amount;

**return** **true**;

}

}

**package** com.zensar;

**public** **class** SavingAccount **extends** BankAccount {

@Override

**public** **void** deposite(**double** amount) {

// **TODO** Auto-generated method stub

**super**.deposite(amount);

}

@Override

**public** **boolean** withdraw(**double** amount) {

// **TODO** Auto-generated method stub

**return** **super**.withdraw(amount);

}

}

**package** com.zensar;

**public** **class** BankDemo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

CurrentAccount c = **new** CurrentAccount();

SavingAccount s = **new** SavingAccount();

c.deposite(500.00);

s.deposite(500.00);

*doWithdrawal*(c); // Withdraw succeed.

*doWithdrawal*(s); // Withdraw failed.

}

**public** **static** **void** doWithdrawal(BankAccount acc) {

**boolean** result = acc.withdraw(1000.00);

**if** (result) {

System.***out***.println("Withdraw succeed.");

} **else** {

System.***out***.println("Withdraw failed.");

}

}

}

Output:

Withdraw succeed.

Withdraw failed.