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.

Employee.java:

public class Employee {  
 private int id;  
 private String name;  
 private String role="Employee";  
 private int baseSalary=1000;  
  
 public Employee(int id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
  
 public String getDetails(){  
 return "Id: "+getId()+  
 " Name: "+name+  
 " Role: "+role+  
 " Total Salary: "+baseSalary;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getName() {  
 return name;  
 }  
}

Manager.java:

public class Manager extends Employee{  
 private int incentive=5000;  
 private int salary=10000;  
 private String role="Manager";  
  
 public Manager(int id,String name){  
 super(id,name);  
 }  
 @Override  
 public String getDetails(){  
 return "Id: "+getId()+  
 " Name: "+getName()+  
 " Role: "+role+  
 " Total Salary: "+(incentive+salary);  
 }  
}

Labour.java:

public class Labour extends Employee{  
 private int overTimeWage=500;  
 private int wage=2000;  
 private String role="Labour";  
  
 public Labour(int id,String name) {  
 super(id,name);  
 }  
 @Override  
 public String getDetails(){  
 return "Id: " +getId()+  
 " Name: "+getName()+  
 " Role: "+role+  
 " Total Salary: "+(overTimeWage+wage);  
 }  
}

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.

BankAccount.java

public class BankAccount {  
 private int accountNumber;  
 private String name;  
 private int currentBalance;  
  
 public BankAccount(int accountNumber, String name,int currentBalance) {  
 this.accountNumber = accountNumber;  
 this.name = name;  
 this.currentBalance=currentBalance;  
 }  
 public void deposit(int deposit){  
 currentBalance+=deposit;  
 System.*out*.println("Amount Deposited, Updated balance = "+currentBalance);  
 }  
 public void withdraw(int withdraw){  
 if(currentBalance<withdraw)  
 System.*out*.println("Insufficient balance");  
 else{  
 currentBalance-=withdraw;  
 System.*out*.println("Amount Withdrawn, Updated balance = "+currentBalance);  
 }  
 }  
 public void getTotalCash(){  
 System.*out*.println("Total Cash = "+currentBalance);  
 }  
  
 public int getAccountNumber() {  
 return accountNumber;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getCurrentBalance() {  
 return currentBalance;  
 }  
}

SavingsAccount.java

public class SavingsAccount extends BankAccount{  
 public SavingsAccount(int accountNumber, String name, int currentBalance) {  
 super(accountNumber, name, currentBalance);  
 }  
 private int fixedDeposit= (int) ((0.5)\*getCurrentBalance());  
 private int currBalance=getCurrentBalance()-fixedDeposit;  
 private String type="Savings";  
 public void setFixedDeposit(int fixedDeposit) {  
 this.fixedDeposit = fixedDeposit;  
 }  
  
 public int getFixedDeposit() {  
 return fixedDeposit;  
 }  
 @Override  
 public void deposit(int deposit){  
 this.fixedDeposit+=deposit;  
 System.*out*.println("Amount deposited, Updated fixed Deposit = "+fixedDeposit);  
 }  
 @Override  
 public void withdraw(int withdraw){  
 if(this.currBalance<withdraw)  
 System.*out*.println("Insufficient balance");  
 else{  
 this.currBalance-=withdraw;  
 System.*out*.println("Amount Withdrawn, Updated balance = "+currBalance);  
 }  
 }  
 @Override  
 public void getTotalCash(){  
 System.*out*.println("Account Type: "+type);  
 System.*out*.println("Total Cash = "+currBalance);  
 System.*out*.println("Fixed Deposit Amount = "+fixedDeposit);  
 }  
  
  
}

CurrentAccount.java

public class CurrentAccount extends BankAccount{  
 public CurrentAccount(int accountNumber, String name, int currentBalance) {  
 super(accountNumber, name, currentBalance);  
 }  
 private int creditLimit=getCurrentBalance();  
 private String type="Current";  
 @Override  
 public void deposit(int deposit){  
 if(creditLimit==getCurrentBalance()){  
 System.*out*.println("Credit balance full cannot deposit any further.");  
 }  
 creditLimit+=deposit;  
 System.*out*.println("Amount Deposited, Updated balance = "+creditLimit);  
 }  
 @Override  
 public void withdraw(int withdraw){  
 if(creditLimit<withdraw)  
 System.*out*.println("Insufficient balance");  
 else{  
 creditLimit-=withdraw;  
 System.*out*.println("Amount Withdrawn, Updated balance = "+creditLimit);  
 }  
 }  
 @Override  
 public void getTotalCash(){  
 System.*out*.println("Account Type: "+type);  
 System.*out*.println("Total Cash = "+creditLimit);  
 }  
}