Harman assignment 2

Q1: Implement following UML diagram, Write a program to test Employee class.

```
package day2;
import java.util.Scanner;
class emp{
	private int id;
	private String name;
	private String surname;
	private int salary;

	public void set()
	{
		Scanner s= new Scanner(System.in);
		System.out.println("enter id,name,surname and salary");
		this.id=s.nextInt();
		this.surname=s.next();
		this.name=s.next();

		this.salary=s.nextInt();
		s.close();

	}
	public void get()
	{

	System.out.println("id:"+id+"\nname:"+name+"\t"+surname+"\nsalary:"+salary)
;
	}

}

public class employee {
	public static void main(String[]args)
	{
		emp e1=new emp();
		e1.set();
		e1.get();
	}

}
```

Q2. Create a book store application which will help a book store to keep the record of its books. For each book, the application will have the Book Title, Book Author, Book ISBN along with the number of copies for each book. The system will allow you to display all books, order new/existing books and sell books. With sell or order of existing books, number of copies will

decrease/increase. With order of new book, a new book entry will be added to the system.

```
package day2;

public class Book {

	private String bookTitle;
	private String author;
	private String isbn;
	private int numOfCopies;

	public String getBookTitle() {
		return bookTitle;
	}

	public String getAuthor() {
		return author;
	}

	public String getIsbn() {
		return isbn;
	}

	public int getNumOfCopies() {
		return numOfCopies;
	}

	public void setNumOfCopies(int numOfCopies) {
		this.numOfCopies = numOfCopies;
	}

	public Book(String bookTitle, String author, String isbn, int numOfCopies)
{
		this.bookTitle = bookTitle;
		this.author = author;
		this.isbn = isbn;
		this.numOfCopies = numOfCopies;
	}

	public void display() {
		System.out.println("book details");
		System.out.println("bookTitle: " + bookTitle);
		System.out.println("author: " + author);
		System.out.println("isbn: " + isbn);
		System.out.println("numOfCopies: " + numOfCopies);
		System.out.println("----------------------");

	}

}


package day2;

public class BookStore {
	private final int SIZE;
	private String bookStoreName;
```

```
	private Book[] books;

	public BookStore(String bookStoreName, int size) {
		SIZE=size;
		this.bookStoreName = bookStoreName;
		this.books = new Book[SIZE];
		init();
	}
	//populate some books
	private void init() {
			books[0]=new Book("rich dad poor dad", "RK", "54545A12", 10);
			books[1]=new Book("basic physics", "abc", "54595A12", 20);
			books[2]=new Book("pointer in c", "YK", "94545A12", 8);
			books[3]=new Book("head first core java", "foo", "50545A12",
20);
			books[4]=new Book("spring in action", "bar", "14545A12", 10);
	}



	public void sell(String bookTitle, int noOfCopies) {
		boolean found=false;
		for(int i=0;i<SIZE; i++) {
			if(	books[i].getBookTitle().equals(bookTitle)) {

				books[i].setNumOfCopies(books[i].getNumOfCopies()-
noOfCopies);
				found=true;
			}
		}

		if(found)
			System.out.println("book sell is successful");
		else
			System.out.println("book is not found in store");
	}

	public void order(String bookTitle, int noOfCopies) {
		boolean found=false;
		for(int i=0;i<SIZE; i++) {
			if(	books[i].getBookTitle().equals(bookTitle)) {

	books[i].setNumOfCopies(books[i].getNumOfCopies()+noOfCopies);
				found=true;
			}
		}
		if(found)
			System.out.println("book order is successful");
		else
			System.out.println("book order is not successful");
	}

	public void display() {
		System.out.println("Book store");
		System.out.println(bookStoreName);
		System.out.println("^^^^^^^^^^^^^^^^Books
details^^^^^^^^^^^^^^^^^^^^^^^^");
		for(Book book: books) {
			book.display();
```

```
            }
            System.out.println("\n\t\t\t");

      }
}
```

```
package day2;



public class Q2 {
      public void main(String[]args)
      {
            BookStore bookStore=new BookStore("harman", 5);
            bookStore.sell("spring in action", 2);
            bookStore.display();
      }


}
```

Assignment on Inheritance, Method overriding, Method overloading

Q3. A Banking System

```
package day2;



      abstract public class Account {
            private String name;
            private String accountNumber;
            private double accountBalance;

            public String getName() {
                  return name;
            }
            public String getAccountNumber() {
                  return accountNumber;
            }

            public double getAccountBalance() {
                  return accountBalance;
            }

            public void setAccountBalance(double accountBalance) {
                  this.accountBalance = accountBalance;
            }

            public Account(String name, String accountNumber, double
accountBalance) {
                  this.name = name;
                  this.accountNumber = accountNumber;
                  this.accountBalance = accountBalance;
            }
```

```
	public void deposit(double amount) {
		accountBalance+=amount;
	}

	public abstract void withdraw(double amount);
	@Override
	public String toString() {
		return "Account [name=" + name + ","
				+ " accountNumber=" + accountNumber + ",
accountBalance=" + accountBalance
				+ "]";
	}


}

package day2;

public class CurrentAccount extends Account {

	private String tradeLicenseNumber;
	private double overdraft;

	public CurrentAccount(String name, String accountNumber, double
accountBalance,
			String tradeLicenseNumber,double overdraft) {
		super(name, accountNumber, accountBalance);
		this.tradeLicenseNumber = tradeLicenseNumber;
		this.overdraft = overdraft;
	}

	public void withdraw(double amount) {
		double allowedWithdrawal= getAccountBalance()+ overdraft;

		if(allowedWithdrawal<=amount) {
			setAccountBalance(getAccountBalance()-amount);
			System.out.println("amount "+ amount+" is withdrawn");
		}else
			System.out.println("You can not withdraw, dont have sufficient
balance and overdraft limit  ...");
	}

	@Override
	public String toString() {
		return super.toString()+
				"CurrentAccount [tradeLicenseNumber=" +
tradeLicenseNumber + ", "
				+ "overdraft=" + overdraft + "]";
	}


}
package day2;

public class SavingsAccount extends Account {
	private double interest=5.0;
	private double maxWithdrawAmountLimit;
	private double minimumBalance;
```

```
	public SavingsAccount(String name, String accountNumber, double
accountBalance,
			double interest,double minimumBalance) {
		super(name, accountNumber, accountBalance);
		this.interest=interest;
		this.minimumBalance=minimumBalance;

		this.maxWithdrawAmountLimit=accountBalance-minimumBalance;
	}

	public double getBalance() {
		return getAccountBalance()*(100+interest)/100;
	}

	public void withdraw(double amount) {
		if(amount<=maxWithdrawAmountLimit) {
			setAccountBalance(getAccountBalance()-amount);
			System.out.println("amount "+ amount+" is withdrawn");
		}else
			System.out.println("You can not withdraw as min balance
required ...");
	}

	@Override
	public String toString() {
		return super.toString()+
				" interest=" + interest + ","
				+ " maxWithdrawAmountLimit=" + maxWithdrawAmountLimit
				+ ", minimumBalance=" + minimumBalance + "]";
	}



}

package day2;

public class Q3 {
public static void main(String[] args) {

		Account account=new
				SavingsAccount("ravi", "4343433434", 30000, 5, 1000);


		Account account2=new
			CurrentAccount("rajiv", "543545445", 500000, "AB1234", 50000);
}}
```

Q4. An Employee Record System

```
package day2;

public class Employe {

	private int id;
```

```
	private String name;

	public Employee(int id, String name) {
		this.id = id;
		this.name = name;
	}
	public int getId() {
		return id;
	}
	public String getName() {
		return name;
	}

	abstract public double getPayment() ;
	@Override
	public String toString() {
		return "Employee [id=" + id + ", name=" + name + "]";
	}

}
```

```
package day2;

public class CommissionEmployee  extends Employee {

	private double commisionPercentage;
	private double totalSales;

	public CommissionEmployee(int id, String name, double commisionPercentage,
double totalSales) {
		super(id, name);
		this.commisionPercentage = commisionPercentage;
		this.totalSales = totalSales;
	}

	@Override
	public double getPayment() {
		return totalSales * commisionPercentage / 100;
	}


}
```

```
package day2;

public class HourlyEmployee  extends Employee {

	private double  hourlyRate;
	private int numberOfHours;

	public HourlyEmployee(int id, String name,double  hourlyRate,
	 int numberOfHours ) {
```

```
		super(id, name);
		this.hourlyRate=hourlyRate;
		this.numberOfHours=numberOfHours;
	}

	@Override
	public String toString() {
		return super.toString()+
				"HourlyEmployee [hourlyRate=" + hourlyRate + ", "
				+ "numberOfHours=" + numberOfHours + "]";
	}

	@Override
	public double getPayment() {
		return hourlyRate*numberOfHours;
	}


}
```

```
package day2;

public class SalariedEmployee  extends Employee {

	private double fixedWeeklySalary;
	public SalariedEmployee(int id, String name, double fixedWeeklySalary) {
		super(id, name);
		this.fixedWeeklySalary=fixedWeeklySalary;
	}
	@Override
	public double getPayment() {
		return fixedWeeklySalary;
	}

}
```

```
package day2;

public class Q4 {


		public static void main(String[] args) {
			Employee employee=new
					CommissionEmployee(121, "amit", 2.1, 2000000);
			System.out.println(employee.getPayment());

	}


}
```

Q5 The Payment System

```
package day2;

public interface payable {
	abstract public double getPayment();
}
```

```
package day2;

public class invoice implements Payable {

	private String partNum;
	private String partDescription;
	private int quantity;
	private double pricePerPart;

	public String getPartNum() {
		return partNum;
	}

	public void setPartNum(String partNum) {
		this.partNum = partNum;
	}

	public String getPartDescription() {
		return partDescription;
	}

	public void setPartDescription(String partDescription) {
		this.partDescription = partDescription;
	}

	public int getQuantity() {
		return quantity;
	}

	public void setQuantity(int quantity) {
		this.quantity = quantity;
	}

	public double getPricePerPart() {
		return pricePerPart;
	}

	public void setPricePerPart(double pricePerPart) {
		this.pricePerPart = pricePerPart;
	}

	public Invoice(String partNum, String partDescription, int quantity, double
pricePerPart) {
		this.partNum = partNum;
		this.partDescription = partDescription;
		this.quantity = quantity;
		this.pricePerPart = pricePerPart;
```

```
	}

	@Override
	public double getPayment() {
		return quantity*pricePerPart;
	}}
```

```
package day2;

public class employees  implements Payable {

		private int id;
		private String name;

		public Employee(int id, String name) {
			this.id = id;
			this.name = name;
		}
		public int getId() {
			return id;
		}
		public String getName() {
			return name;
		}

		abstract public double getPayment() ;
		@Override
		public String toString() {
			return "Employee [id=" + id + ", name=" + name + "]";
		}


	}
```

```
package day2;

public class Q5 {
	public static void main(String[] args) {
		Employee employee=new
				CommissionEmployee(121, "amit", 2.1, 2000000);
		Invoice invoice=new Invoice("A123", "Electronic23W", 20, 3000);

		PaymentProcessingSystem.processPayment(employee);
	}

}
```