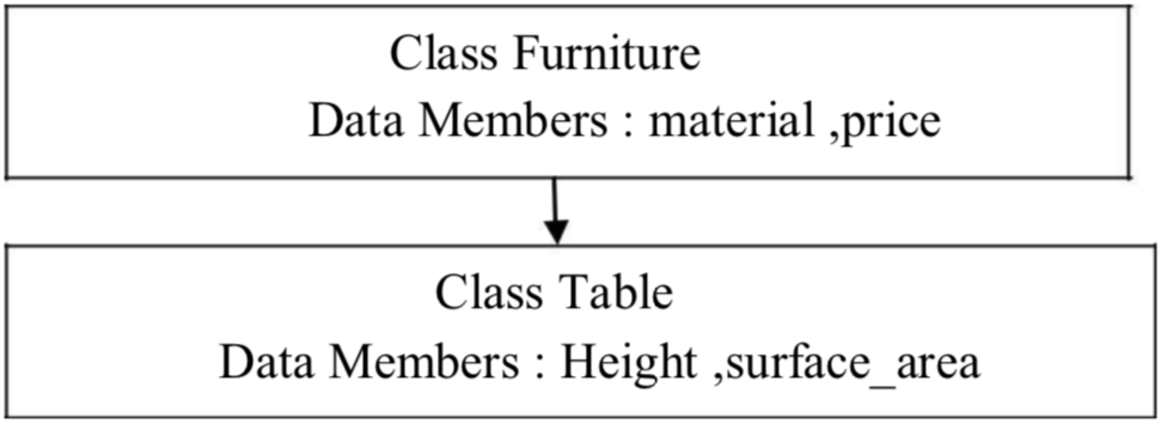
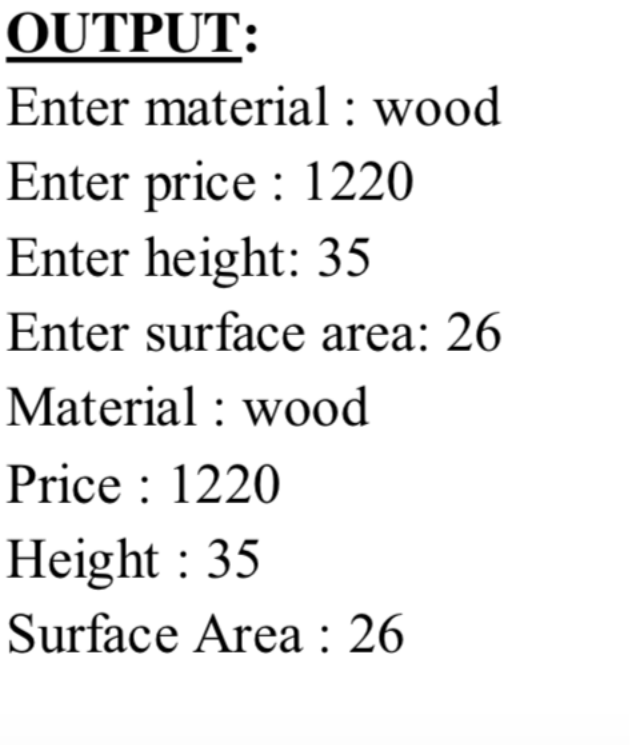
Lab Task 3

1. Write a program to implement single inheritance from following figure. Accept and display data for one table.

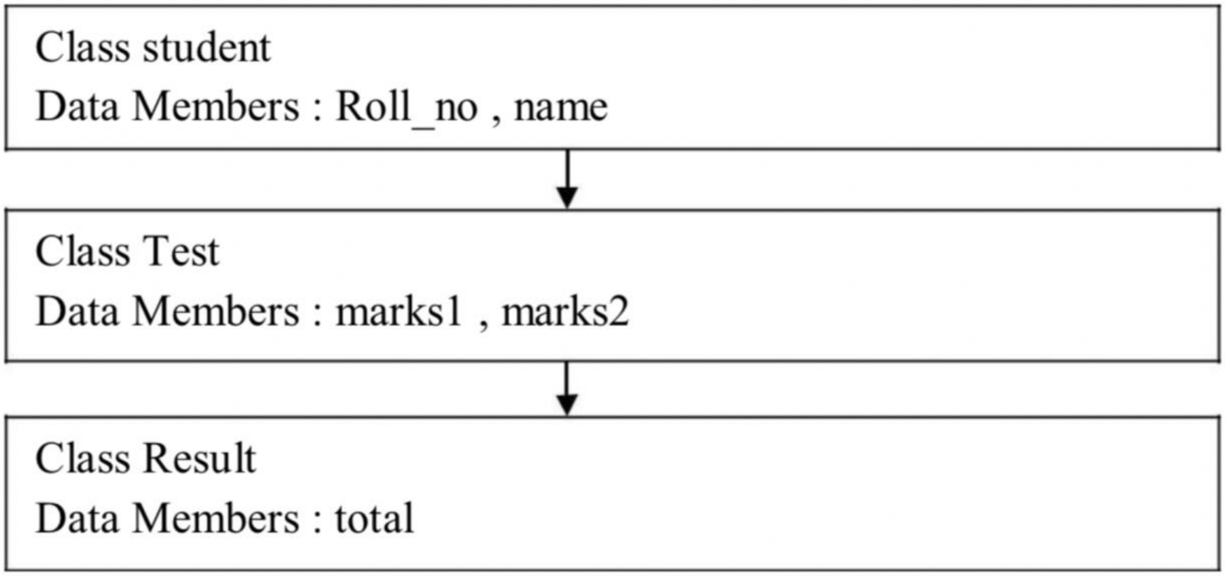

Class Furniture
Data Members : material ,price
Class Table
Data Members : Height ,surface_area



OUTPUT:

```
Enter material : wood
Enter price : 1220
Enter height: 35
Enter surface area: 26
Material : wood
Price : 1220
Height : 35
Surface Area : 26
```

2. Write a program to implement multilevel inheritance from following figure. Accept and display data for one student.



Class student
Data Members : Roll_no , name

Class Test
Data Members : marks1 , marks2

Class Result
Data Members : total

Output

```
Roll no: 101

Name: Anders Hejlsberg

Marks1: 100

Marks2: 100

Total: 200
```

3. a.

Create an application named **JobDemo** that declares and uses Job objects. The Job class holds job information for a home repair service. The class has five properties that include a job number, customer name, job description, estimated hours, and price for the job. Create a constructor that requires parameters for all the data except price. Include auto-implemented properties for the job number, customer name, and job description, but not for hours or price; the price field value is calculated as estimated hours times $45.00 whenever the hours value is set. Also create the following for the class:

- An Equals() method that determines two Jobs are equal if they have the same job number
- A GetHashCode() method that returns the job number
- A ToString() method that returns a string containing all job information

The JobDemo application declares a few Job objects, sets their values, and demonstrates that all the methods work as expected.

b. Using the Job class you created in Exercise 3a, write a new application named **JobDemo2** that creates an array of five Job objects. Prompt the user for values for each Job. Do not allow duplicate job numbers; force the user to reenter the job when a duplicate job number is entered. When five valid objects have been entered, display them all, plus a total of all prices.

c. Create a RushJob class that derives from Job. A RushJob has a $150.00 premium that is added to the normal price of the job. Override any methods in the parent class as necessary. Write a new application named **JobDemo3** that creates an array of five RushJobs. Prompt the user for values for each, and do not allow duplicate job numbers. When five valid RushJob objects have been entered, display them all, plus a total of all prices.

d. Make any necessary modifications to the RushJob class so that it can be sorted by job number. Modify the JobDemo3 application so the displayed orders have been sorted. Save the application as **JobDemo4**.