# Lab Session 06: Apply the concepts of Method Overloading and Method Overriding

**Date of the Session: Time of the Session: \_\_\_\_\_to\_\_\_\_\_\_**

**Pre Lab Tasks:**

**Pre-Lab Tasks:**

**1. Is it possible to define two or more methods within the same class that share the same name?**

**2. When java will employ its automatic type conversion in method overloading scenario?**

**3. When the method in the subclass is said to override the method in the superclass?**

**4. Give an example of Method Overriding.**

**In-Lab Tasks:**

1. Write a program to find the maximum of given numbers. In first case, have a max method with two integer parameters. In second case, have a max method with two double parameters. In third case, have a max method with three double parameters.

**Test Case:**

***Test Case 1:***

***Input:***

2,6

2.4, 6.4

2.4,8.6,6.4

***Output:***

6

6.4

8.6

**Program:**

class Main{

static int max(int a, int b){

return a>b?a:b;

}

static float max(float a,float b){

return a>b?a:b;

}

static double max(double a,double b){

return a>b?a:b;

}

public static void main(String [] args){

System.out.println("max:45,54="+max(45,54) );

System.out.println("max:457.4646,54.56="+max(457.4646,54.56) );

System.out.println("max:45.0457438453,196.27352935="+max(45.0457438453,196.27352935) );

}

}

**OUTPUT:**

****

1. Create a class called book with attributes name, author and methods setdata() and display() to read and display values of attributes. Extend book class in to other two sub classes with the names McGraw-Hill and Oxford. Include new price() method in both sub classes to specify price of books in that publication.

**Test Cases:**

***Test Case 1:***

***Input:***

“Java Programming”, “Robert”, 300, 400

***Output:***

Name= Java Programming

Author= Robert

McGraw-Hill Price=300

Oxford Price=400

**Program:**

import java.util.\*;

class Book{

String name ="";

String author = "";

public void setName(String name){

this.name = name;

}

public void setAuthor(String author){

this.author = author;

}

public void display(){

System.out.println("Name:"+name+"\nAuthor:"+author);

}

}

class Oxford extends Book{

int price;

public void setPrice(int price){

this.price = price;

}

public void display(){

super.display();

System.out.println("Oxford-Price:"+price);

}

}

class McGrawHill extends Book{

int price;

public void setPrice(int price){

this.price = price;

}

public void display(){

super.display();

System.out.println("McGraw-Hill-Price:"+price);

}

}

class Main{

public static void main(String [] args){

Oxford ox = new Oxford();

McGrawHill mc = new McGrawHill();

System.out.println("Enter the details of the Book\n(Name, Author, McGraw-Hill-Price, Oxford-Price)");

Scanner sc= new Scanner(System.in);

String name = sc.nextLine();

String author = sc.nextLine();

int oxprice = sc.nextInt();

int mcprice = sc.nextInt();

ox.setName(name);

ox.setAuthor(author);

ox.setPrice(oxprice);

mc.setName(name);

mc.setAuthor(author);

mc.setPrice(mcprice);

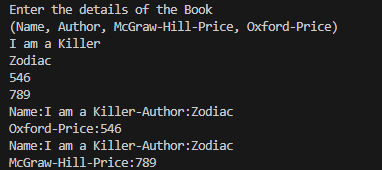
ox.display();

mc.display();

}

}

**OUTPUT:**



**Post-Lab Tasks:**

1. Create a class called IPLcricket with methods played\_matches(), win() and loss(). Extend this class into other two sub classes with the names Hyderabad\_sunrises and Chennai\_superkings. Display number of matches win and loss by each team.

**Test Cases:**

***Test Case 1:***

***Input:***

4,2,2,3,2,1

***Output:***

Hyderabad\_sunrisesTeam summary:

Played Matches =4

Won = 2

Loss= 2

Chennai\_superkingsTeam summary:

Played Matches =3

Won = 2

Loss= 1

***Test Case 1:***

***Input:***

3,2,2, 3,2,1

***Output:***

Hyderabad\_sunrisesTeam summary:

Invalid win and loss data

Chennai\_superkingsTeam summary:

Played Matches =3

Won = 2

Loss= 1

**Students Signature**

|  |  |  |
| --- | --- | --- |
| *(For Evaluator’s use only)* | | |
|  | Comment of the Evaluator (if Any) | Evaluator’s Observation  Marks Secured: \_\_\_\_\_\_\_ out of \_\_\_\_\_\_\_\_  Full Name of the Evaluator:  Signature of the Evaluator Date of Evaluation: |
|  |  |