**LECTURE 7-8: NOTES**

**My notes:**

1. Instantiation in Java
2. Constructors
3. OOP Principles
4. Inheritance
5. Single Inheritance
6. Multilevel Inheritance
7. Hierarchical Inheritance
8. Method Overriding
9. Constructor in inheritance
10. [Polymorphism](https://www.scaler.com/topics/java/polymorphism-in-java/)
11. Method Overriding
12. Method Overloading

**Practice Problems**

1. Write a Java program to create a class called Animal with a method called makeSound(). Create a subclass called Cat that overrides the makeSound() method to bark.
2. Write a Java program to create a class called Shape with a method called getArea(). Create a subclass called Rectangle that overrides the getArea() method to calculate the area of a rectangle.

**Sample Example of Inheritance:**

**class** Shape {

**public** **double** getArea() {

**return** 0.0;

}

}

**class** Rectangle **extends** Shape {

**private** **double** length;

**private** **double** width;

**public** Rectangle(**double** length, **double** width) {

**this**.length = length;

**this**.width = width;

}

@Override

**public** **double** getArea() {

**return** length \* width;

}

}

**public** **class** Main {

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

Rectangle rectangle = **new** Rectangle(3.0, 10.0);

**double** area = rectangle.getArea();

System.***out***.println("The area of the rectangle is: " + area);

}

}

**Sample Example of Inheritance:**

**class** shape{

**int** side;

**public** **static** **void** area(**int** l)

{

}

}

**class** square **extends** shape{

**public** **static** **void** area(**int** l)

{

System.***out***.println("Area of square is"+l\*l);

}

}

**class** rectangle **extends** shape{

**int** side2;

**public** **static** **void** area(**int** l,**int** b)

{

System.***out***.println("Area of square is"+l\*b);

}

}

**public** **class** Main{

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

{

square obj=**new** square();

obj.side=20;

obj.*area*(obj.side);

rectangle obj1=**new** rectangle();

obj1.side=20;

obj1.side2=10;

obj1.*area*(obj1.side,obj1.side2);

}

}

Example Covered in class:

**import** java.util.Scanner;

**class** BasicCal{

**int** num1;

**int** num2;

**public** **static** **void** add(**int** n1,**int** n2)

{

**int** sum=n1+n2;

System.***out***.println("sum"+sum);

}

}

**class** ComplexCal **extends** BasicCal{

**public** **static** **void** sub(**int** n1,**int** n2)

{

**int** sub=n1-n2;

System.***out***.println("sub is"+sub);

}

}

**class** Main{

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

{

ComplexCal obj=**new** ComplexCal();

Scanner n=**new** Scanner(System.***in***);

**int** num1=n.nextInt();

**int** num2=n.nextInt();

obj.*add*(num1, num2);

obj.*sub*(num1,num2);

}

}

Example covered in class 2c:

**class** BasicCal{

**int** num1;

**int** num2;

**public** **static** **void** add(**int** num1,**int** num2)

{

**int** sum=num1+num2;

System.***out***.println("sum"+sum);

}

**public** **void** display()

{

System.***out***.println("We are doing only addition");

}

}

**class** ComplexCal **extends** BasicCal{

**public** **static** **void** sub(**int** num1,**int** num2)

{

**int** sub=num1-num2;

System.***out***.println("sub"+sub);

}

**public** **void** display()

{

System.***out***.println("We are doing addition and subtraction");

}

}

**class** SuperComplexCal **extends** ComplexCal{

**public** **void** mul(**int** n1,**int** n2)

{

**int** mul=n1\*n2;

System.***out***.println("mul"+mul);

}

}

**public** **class** Main {

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

{

BasicCal obj=**new** BasicCal();

obj.num1=20;

obj.num2=30;

obj.*add*(obj.num1, obj.num2);

obj.display();

}

}