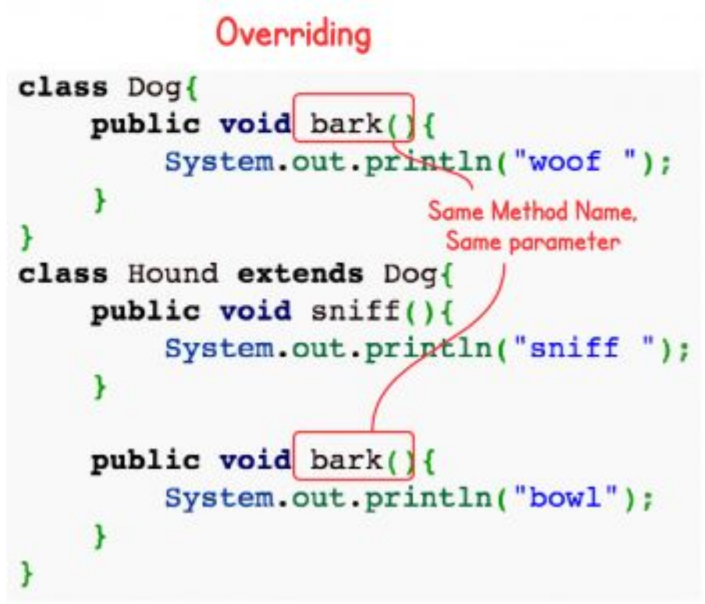
# CSC 110 2.0
# Object Oriented Programming

Tute 03

Polymorphism, Inheritance

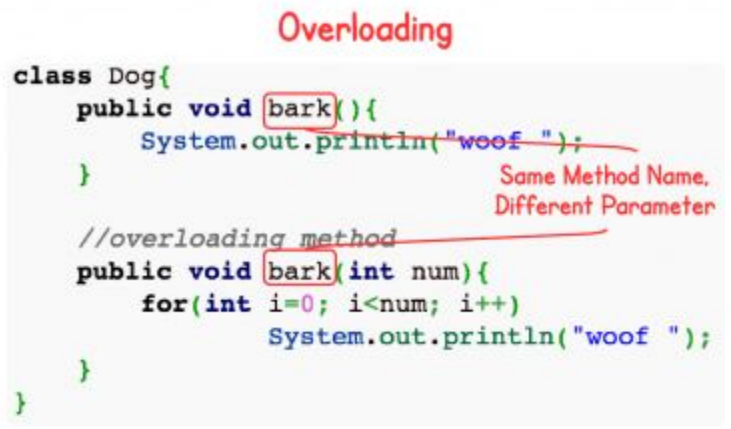
Dulan S. Dias © 2019

# Polymorphism (Overriding and Overloading)



```java
class Dog{
    public void bark(){
        System.out.println("woof ");
    }
}
class Hound extends Dog{
    public void sniff(){
        System.out.println("sniff ");
    }

    public void bark(){
        System.out.println("bowl");
    }
}
```



```java
class Dog{
    public void bark(){
        System.out.println("woof ");
    }

    //overloading method
    public void bark(int num){
        for(int i=0; i<num; i++)
            System.out.println("woof ");
    }
}
```

## Example - Overriding

```
class Complex {
    private double re, im;
    public Complex(double re, double im) {
        this.re = re;
        this.im = im;
    }
}
```

## Example - Overriding (contd.)

```
// Driver class to test the Complex class
public class Main {
    public static void main(String[] args) {
        Complex c1 = new Complex(10, 15);
        System.out.println(c1);
    }
}
```

## Example - Overriding (contd.)

```
// Driver class to test the Complex class
public class Main {
    public static void main(String[] args) {
        Complex c1 = new Complex(10, 15);
        System.out.println(c1);
    }
}
```

```
Complex@19821f
```

## Example - Overriding (contd.)

```
class Complex {
    private double re, im;

    public Complex(double re, double im) {
        this.re = re;
        this.im = im;
    }

    @Override
    public String toString() {
        return (re + " + i " + im);
    }
}
```

## Example - Overriding (contd.)

```
// Driver class to test the Complex class
public class Main {
    public static void main(String[] args) {
        Complex c1 = new Complex(10, 15);
        System.out.println(c1);
    }
}
```

## Example - Overriding (contd.)

```
// Driver class to test the Complex class
public class Main {
    public static void main(String[] args) {
        Complex c1 = new Complex(10, 15);
        System.out.println(c1);
    }
}
```

```
10 + i 15
```

## Q1

Create a class '**Degree**' having a method '**getDegree**' that prints "I got a degree". It has two subclasses namely '**Undergraduate**' and '**Postgraduate**' each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively.

Call the method by creating an object of each of the three classes.

## Q2

A class has an integer data member '**i**' and a method named '**printNum**' to print the value of '**i**'. Its subclass also has an integer data member '**j**' and a method named '**printNum**' to print the value of '**j**'. Make an object of the subclass and use it to assign a value to '**i**' and to '**j**'.

Now call the method '**printNum**' by this object.

## Inheritance

Two concepts:

- subclass (child) - the class that inherits from another class
- superclass (parent) - the class being inherited from

To inherit from a class, use the `extends` keyword.

## Q1

Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call:

1 - method of parent class by object of parent class

2 - method of child class by object of child class

3 - method of parent class by object of child class

## Q2

Create a class named 'Member' having the following members:

Data members

1 - Name

2 - Age

3 - Phone number

4 - Address

5 - Salary

## Q2 (contd.)

It also has a method named 'printSalary' which prints the salary of the members.

Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

## Q3

Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'.

Print the area and perimeter of a rectangle and a square.

## Q4

Now repeat the above example to print the area of 10 squares.

Hint - Use array of objects

## Q5

Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle".

Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.