Java Programming I

Session 4

Program Control Statements

Juan Carlos Moreno - UCLA Ex

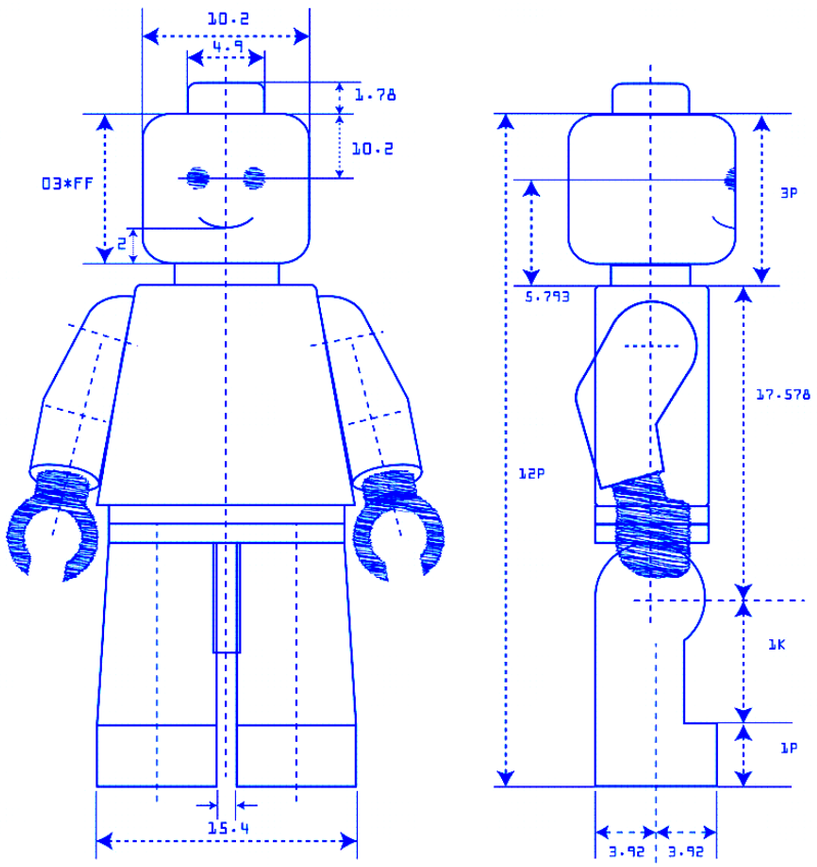
Agenda

- **Classes**
- **Objects**
- **Methods**
- **Constructors**
- **Method overloading**
- **Packages**

Classes

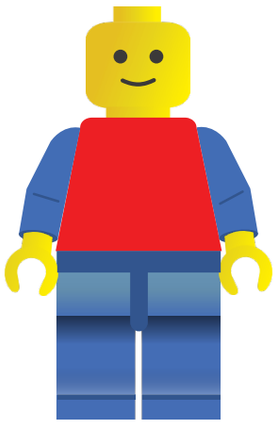
Show me all the blueprints

public class LegoMan

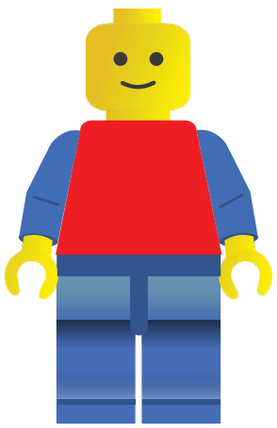


Class

new LegoMan();



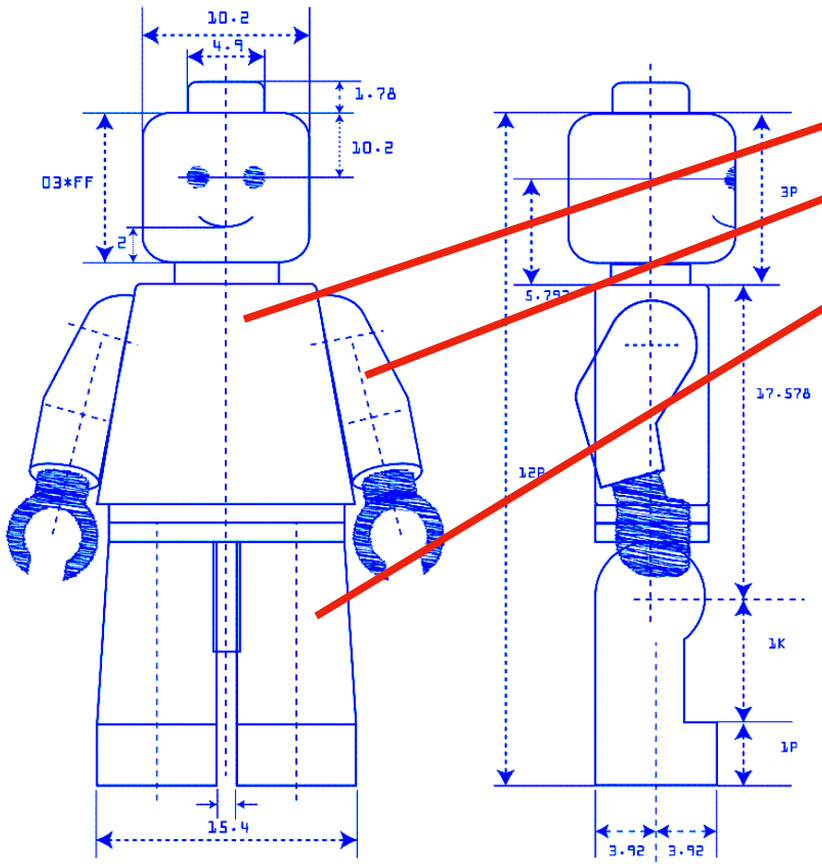
new LegoMan();



Object

Classes

Show me all the blueprints

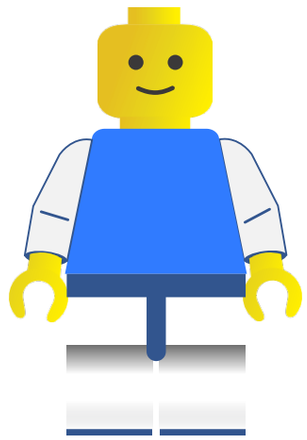


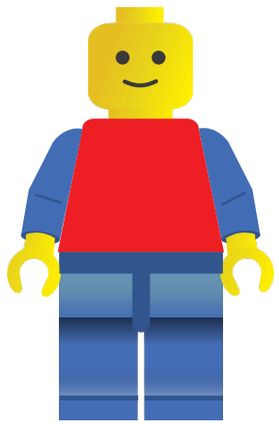

```
public class LegoMan{
    private String shirtColor = "blue";
    private String armColor = “white";
    private String pantsColor = “white”;

    public LegoMan(String shirtColor,
                   String pantsColor,
                   String armColor)
    {
        this.pantsColor = pantsColor;
        this.shirtColor = shirtColor;
        this.armColor = armColor;
    }

}
```

Class to Object

Building Objects

```
LegoMan lm1 = new LegoMan();


LegoMan lm = new LegoMan("red", "blue", “red");
```





Class structure

members, methods

```
public class KlassName{
    public int member1;
    private boolean member2 = true; // Default value override
    private static int classMember;

    public KlassName(){
        // Constructor
        member1 = (int)(Math.random()*100);
    }

    public boolean noArgumentMethod(){
        return this.member2;
    }

    public int argumentMethod(int x){
        return x + member1;
    }

    public static int classMethod(){
        return classMember;
    }
}
```

Could you?

members, methods

```java
public class KlassName{
    public int member1;
    private boolean member2 = true;
    private static int classMember;

    public KlassName(){
        // Constructor
        member1 = 10;
    }

    public boolean noArgumentMethod(){
        return this.member2;
    }

    public int argumentMethod(int x){
        return x + member1;
    }

    public static int classMethod(){
        return classMember;
    }
}
```

```java
KlassName.member1 = 10;

KlassName kn = new KlassName();
System.out.println(kn.member1);

KlassName.classMember = 10;

System.out.println(KlassName.classMethod());

kn.member1 = 10;
```

All together now

A Singleton

Just one and one only

```
public class Singleton{

}
```

Back to 1st class

Object oriented Programming

- Objects and Classes
- Encapsulation - (private, protected, public)
- Composition, Inheritance and Delegation
- Polymorphism

Encapsulation

hiding the implementation details from users

```
public class EncapsulationDemo{
    private int age;

    public int getAge(){
        return age;
    }

    public void setAge(int newValue){
        age = newValue;
    }
}

public class EncapsTest{
    public static void main(String args[]){
        EncapsulationDemo obj = new EncapsulationDemo();
        obj.setAge(32);
        System.out.println("Employee Age: " + obj.getAge());
    }
}
```

Inheritance

Allowing a class to inherit properties and methods from other classes

```
class Vehicle {
    String color;
    int speed;
    int size;

    void attributes() {
        System.out.println("Color : " + color);
        System.out.println("Speed : " + speed);
        System.out.println("Size : " + size);
    }
}

class Car extends Vehicle {
    int CC;
    int gears;

    void attributescar() {
        // The subclass refers to the members of the superclass
        this.attributes();
        System.out.println("CC of Car : " + CC);
        System.out.println("No of gears of Car : " + gears);
    }
}
```

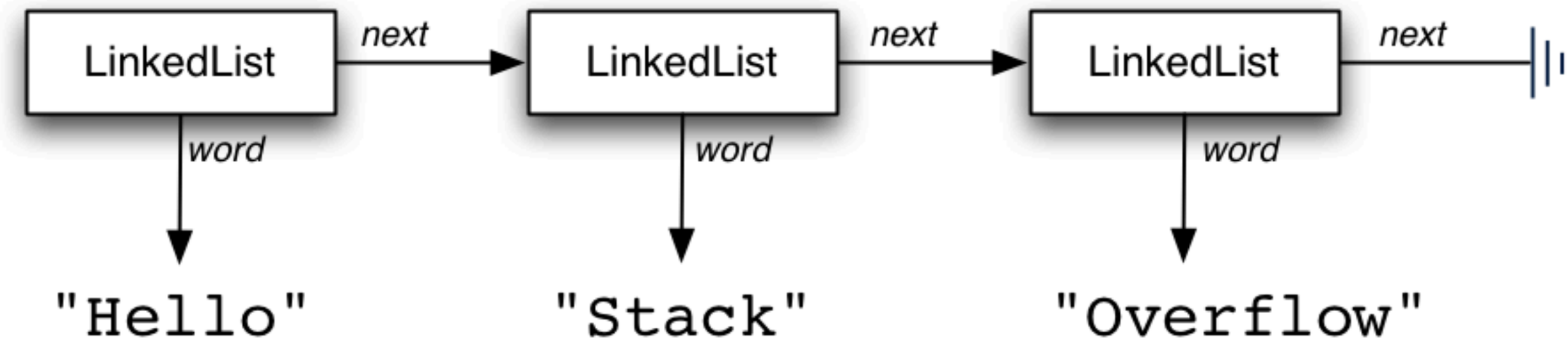
Polymorphism

capability of a method to do different things based on the object that it is acting upon

```
class Overload
{
    void demo (int a)
    {
        System.out.println ("a: " + a);
    }

    void demo (int a, int b)
    {
        System.out.println ("a and b: " + a + "," + b);
    }

    double demo(double a) {
        System.out.println("double a: " + a);
        return a*a;
    }
}
```

LinkedLists

Element knows its value and it knows about the next,



All together now

LinkedList

Implement a linked list in Java

```
class LinkedList
{

}
```

Packages

A way to store objects