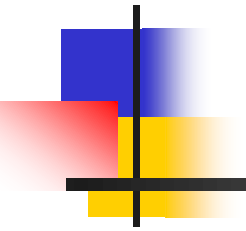


Network Programming

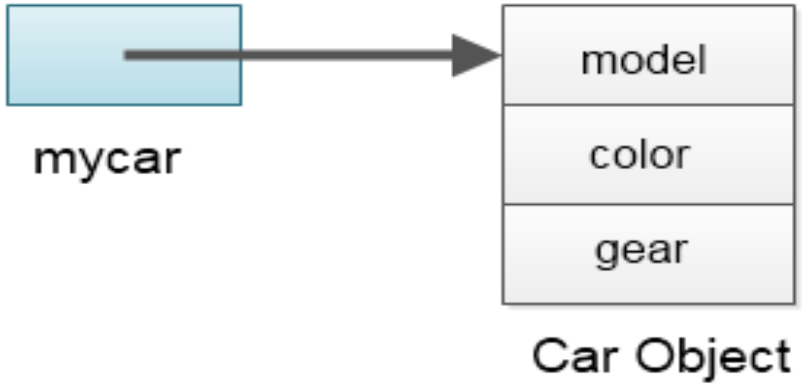
Java – 1

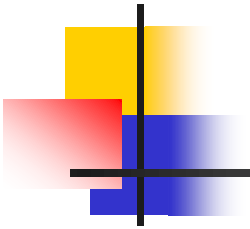
Practice Programming

Java Review (Declaring Objects)

```
class Car{
    String model;
    String color;
    String gear;
    }
  . . .
Car mycar;
```


```
Car mycar = new Car();
    . . .
Car mycar;              // declares
mycar = new Car(); // defines
```

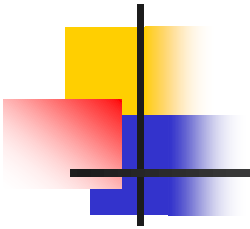




Java Review (this)

```
class Bclass {
    int n;
    void setValue (int n) {this.n = n; }
}

public class AClass {
    public static void main(String[] args) {
        Bclass x = new Bclass();
        x.setValue(3);
        System.out.println( x.n );
    }
}
```

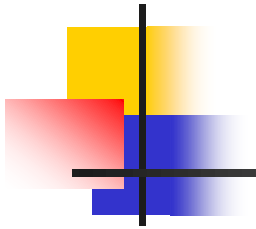


Java Review (this)

```
class Value {
    int x = 1;
    Value me() {return this;}
}

public class Aclass {
    public static void main(String[] args) {
        Value n = new Value();
        System.out.println( n.x );
        System.out.println( n.me().x );              // same as above
        System.out.println( n.me().me().x );         // same as above
    }
}
```

Java Review (this)



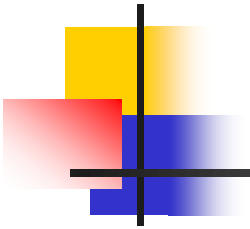
```
class B {
    int n;
    void setMe (int m) {
        C h = new C( );
        h.setValue(this, m);
    }
}

class C {
    void setValue (B obj, int h) {
        obj.n = h;
    }
}
```

```
public class A {
    public static void main(String[] args) {
        B x = new B( );
        x.setMe(3);
        System.out.println( x.n );
    }
}
```

Java Review (super)

```
class Vehicle {
        int speed =50;
}
class Bike extends Vehicle {
        int speed =100;
        void display() {
                System.out.println(speed);
        }
        public static void main ( String args[ ] ){
                Bike b = new Bike();
                b.display();
        }
}
```



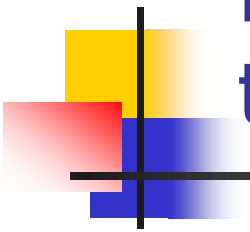
Java Review (super)

```
class Vehicle {
        int speed =50;
}
class Bike extends Vehicle {
        int speed =100;
        void display() {
                System.out.println(super.speed);
        }
        public static void main( String args[]){
                Bike b = new Bike();
                b.display();
        }
}
```

Java Review (super)

```
class Vehicle {
        Vehicle ( ){
                System.out.println("Vehicle is Created");
        }
}
class Bike extends Vehicle {
        Bike( ) {
                super();
                System.out. println ("Bike is Created");
        }
        public static void main( String args[]){
                Bike b = new Bike();
        }
}
```

Four Levels of Access to a Class or object's members



- **private:** accessible from within this class only
  - should be standard practice on instance fields to support OO encapsulation
- **protected:** accessible from any subclass or any class in the package
- **public:** accessible from any class anywhere
- **none** (if you don't specify): accessible from any class in the package

Access Control Levels

access modifier	**class**	**subclass**	**package**	**world**
private	x			
default	x		x	
protected	x	x	x	
public	x	x	x	x

Private

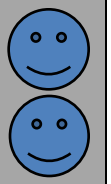
```
class Alpha {
    private int iamprivate;
    private void privateMethod()
     {
            System.out.println
                    ("privateMethod");
    }
}
```

```
class Beta {
    void accessMethod() {
        Alpha a = new Alpha();
        a.iamprivate = 10;
        a.privateMethod();
    }
}
```

Protected

```
package Greek;
class Alpha {
    protected int iamprotected;
    protected void
     protectedMethod() {
        System.out.println
        ("protectedMethod");
    }
}
```

```
package Greek;
class Gamma {
    void accessMethod() {
        Alpha a = new Alpha();
        a.iamprotected = 10;
        a.protectedMethod();
    }
}
```



Protected (II)

```
import Greek.*;
package Latin;
class Delta extends Alpha {
   void accessMethod(Alpha a, Delta d) {
         a.iamprotected = 10;
         d.iamprotected = 10;
         a.protectedMethod();
         d.protectedMethod();
   }
}
```




Public

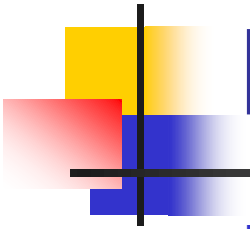
```
package Greek;
public class Alpha {
    public int iampublic;
    public void publicMethod() {
    System.out.println
            ("publicMethod");
    }
}
```

```
import Greek.*;
package Roman;
class Beta {
    void accessMethod() {
            Alpha a = new Alpha();
            a.iampublic = 10;
            a.publicMethod();
    }
}
```

Package

```
package Greek;
class Alpha {
    int iampackage;
    void packageMethod() {
      System.out.println
     ("packageMethod");
    }
}
```

```
package Greek;
class Beta {
    void accessMethod() {
            Alpha a = new Alpha();
            a.iampackage = 10;
            a.packageMethod();
    }
}
```



References

- **Java Network Programming, 4th Edition** (Book)
  - Developing Networked Applications By Elliotte Rusty Harold
  - Publisher: O'Reilly Media Release Date: October 2013
- *www.anwar.agtab.com/Web/**books**/**book**2.pdf*

- **Java 101: Classes and objects in Java**
- http://www.javaworld.com/article/2979739/learn-java/java-101-classes-and-objects-in-java.html

- **Inheritance in Java**
- http://www.javatpoint.com/inheritance-in-java

- **Access modifiers in Java**
- http://o7planning.org/en/10319/access-modifiers-in-java