



© A+ Computer Science - www.apluscompsci.com

Pieces of the OOP Puzzle Review

© A+ Computer Science - www.apluscompsci.com

constructors

```
public Triangle()  
{  
    setSides(0,0,0);  
}
```

**Default
Constructor**

**Constructors are similar to methods.
Constructors set the properties of
an object to an initial state.**

© A+ Computer Science - www.apluscompsci.com

Constructors are typically used to initialize all of the Object's data/properties. The default constructor usually sets all data/properties to a zero value. The exact zero value depends on the specified type of each instance variable.

constructors

```
public Triangle(int a, int b, int c)
{
    setSides(a,b,c);
}
```

**Initialization
Constructor**

**Constructors are similar to methods.
Constructors set the properties of
an object to an initial state.**

© A+ Computer Science - www.apluscompsci.com

Constructors are typically used to initialize all of the Object's data/properties. The initialization constructor usually sets the data/properties to a provided parameter value.

modifier methods

```
public void setSides(int a, int b, int c)
{
    setSideA(a);
    //more of the same
}
```

Modifier methods are methods that change the properties of an object.

© A+ Computer Science - www.apluscompsci.com

Modifier methods are used to modify the Object's data/properties. Set methods are modifier methods.

modifier methods

```
public void setSideA(int a)
{
    sideA=a;
}
```

Modifier methods are methods that change the properties of an object.

© A+ Computer Science - www.apluscompsci.com

Set methods are modifier methods. Set methods are used to change the data/properties of an Object.

accessor methods

```
public int getSideA()  
{  
    return sideA;  
}
```

Accessor methods are methods that retrieve or grant access to the properties of an object, but do not make any changes.

© A+ Computer Science - www.apluscompsci.com

Accessor methods are used to retrieve the data/properties from the Object. Get methods are accessor methods. Accessor methods do not make changes to the data/properties.

accessor methods

```
public String toString()  
{  
    return "" + getSideA() + //more get calls  
}
```

Accessor methods are methods that retrieve or grant access to the properties of an object, but do not make any changes.

© A+ Computer Science - www.apluscompsci.com

The toString() method is an accessor method. The toString() method should return all data/properties. The toString() should not change the data/properties of the Object.

encapsulation

All data members should have private access. The public constructors, accessor methods, and modifier methods should be used to manipulate the data. All data is tucked away nicely inside the class.

© A+ Computer Science - www.apluscompsci.com

encapsulation

The public methods give you access to an object's private data / properties.

**Class/
Object**

private data /
instance variables /
properties

getIt()

setIt()

toString()

© A+ Computer Science - www.apluscompsci.com

**Open
triangle.java
trianglerunner.java**

© A+ Computer Science - www.apluscompsci.com

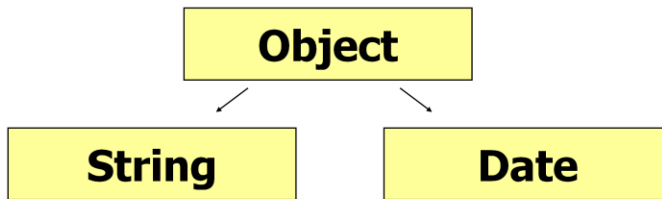
Object



© A+ Computer Science - www.apluscompsci.com

class Object

In Java, all classes are sub classes of class Object. This adds greater flexibility when writing programs in Java.



© A+ Computer Science - www.apluscompsci.com

All classes extend Object!

```
public class Monster extends Object
{
    public void print( )
    {
        out.println("Monster");
    }
}
```

© A+ Computer Science - www.apluscompsci.com

Java automatically applies the extends Object to all new classes created. All classes extend Object.

class Object

**Because all classes are sub
classes of Object, all classes
start with the same methods.**

**.equals()
.toString()
. and more**

© A+ Computer Science - www.apluscompsci.com

Object contains quite a few methods, but equals() and toString() are the most commonly used.

Open monsterone.java

© A+ Computer Science - www.apluscompsci.com



What does public mean?

All members with public access can be accessed inside and outside of the class where they are defined.

© A+ Computer Science - www.apluscompsci.com

What does private mean?

All members with private access can only be accessed inside of the class where they are defined.

© A+ Computer Science - www.apluscompsci.com

Open private.java

© A+ Computer Science - www.apluscompsci.com

Constructors

© A+ Computer Science - www.apluscompsci.com

Constructors

If you do not provide any constructors, Java will provide a default constructor.

© A+ Computer Science - www.apluscompsci.com

Open monstertwo.java

© A+ Computer Science - www.apluscompsci.com

equals

© A+ Computer Science - www.apluscompsci.com

The equals() method

The equals() method is used to see if two objects have the same contents.

```
String one = "comp";  
String two = "sci";  
out.println(one.equals(two));
```

© A+ Computer Science - www.apluscompsci.com

To determine if two Objects are the same, the data/properties of both Objects must be compared.

```

class Monster
{
    private int height;

    //methods

    public boolean equals(Object obj){
        Monster other = (Monster)obj;
        if(getHeight()==other.getHeight())
            return true;
        return false;
    }

    //methods
}

//test code in the main
Monster one = new Monster(33);
Monster two = new Monster(12);
out.println(one.equals(two));

```

equals()

OUTPUT
false

© A+ Computer Science - www.apluscompsci.com

To determine if two Objects are the same, the data/properties of both Objects must be compared.

Monster contains a height property only. Monsters are compared by comparing the heights. If the heights of two Monsters are the same, the Monsters are considered equal.

Open equals.java

© A+ Computer Science - www.apluscompsci.com

Overloading

Overloading occurs when you have more than one method or constructor with the same name. Each method or constructor must have a different parameter list.

of parameters && data types matter

© A+ Computer Science - www.apluscompsci.com

```

class Monster{
    private int height;           //default assigned to 0
    private double weight;       //default assigned to 0

    public Monster(){
        height=0;
        weight=0.0;
    }

    public Monster(int ht){
        height=ht;
        weight=0.0;
    }

    public Monster(double wt){
        height=0;
        weight=wt;
    }

    public Monster(int ht, double wt){
        height=ht;
        weight=wt;
    }
}

```

Overloading

The Monster constructor has been overloaded as it appears 4 times. Each time Monster() is written, a different set of parameters is provided. Java can differentiate between the Monster() constructors by the parameter list.

Open overload.java

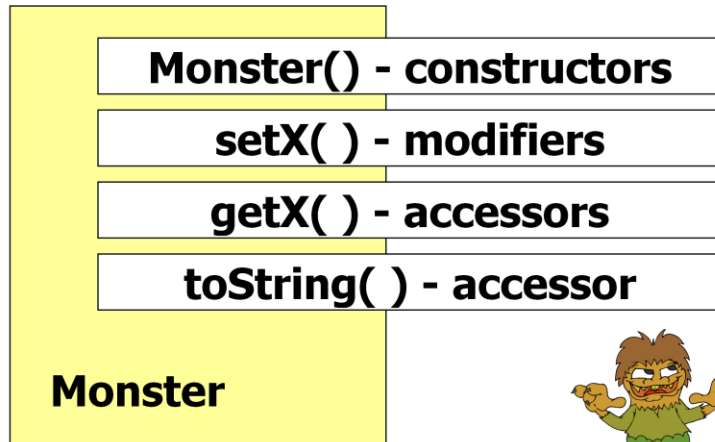
© A+ Computer Science - www.apluscompsci.com

The Monster Class



© A+ Computer Science - www.apluscompsci.com

Monster Object Diagram



© A+ Computer Science - www.apluscompsci.com


```

class Monster
{
    //instance vars / data fields

    public Monster(){ ← constructor
        //code
    }

    public void setX( params ){ ← modifier
        //code
    }

    public int getX(){ ← accessor
        //code
    }

    public String toString() { ← accessor
        //code
    }
}

```

© A+ Computer Science - www.apluscompsci.com

Constructors

```
class Monster{  
    // instance variables  
    public Monster(){ code }  
    public Monster( int ht ) { code }  
    public Monster(int ht, int wt)  
    { code }  
    public Monster(int ht, int wt, int age)  
    { code }  
    //more methods  
}
```

© A+ Computer Science - www.apluscompsci.com

Monster Instantiation 1

Monster m = new Monster();

m



MONSTER

Properties

– height – 0 weight - 0 age - 0

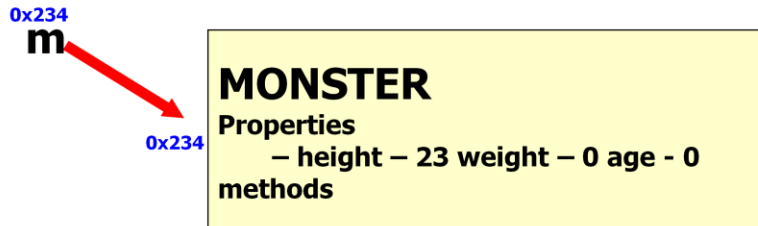
methods

m is a reference variable that refers to a Monster object.

© A+ Computer Science - www.apluscompsci.com

Monster Instantiation 2

Monster m = **new** Monster(23);

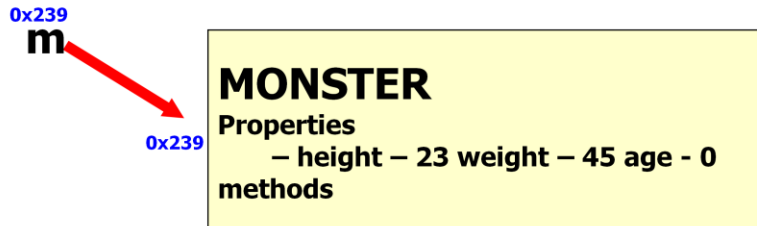


m is a reference variable that refers to a Monster object.

© A+ Computer Science - www.apluscompsci.com

Monster Instantiation 3

Monster m = **new** Monster(23, 45);

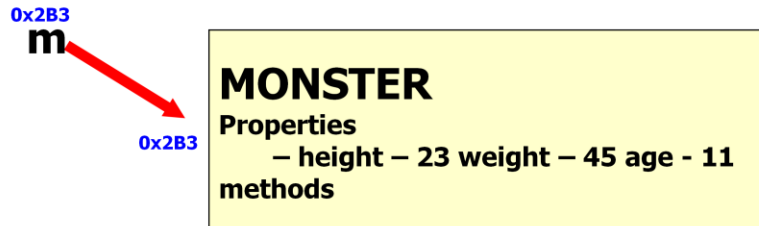


m is a reference variable that refers to a Monster object.

© A+ Computer Science - www.apluscompsci.com

Monster Instantiation 4

Monster m = new Monster(23, 45, 11);



m is a reference variable that refers to a Monster object.

© A+ Computer Science - www.apluscompsci.com

Start work on the labs

© A+ Computer Science - www.apluscompsci.com