Lab 18

Pieces of the OOP Puzzle Review

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.

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.

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.

modifier methods

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

Modifier methods are methods that change the 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.

accessor methods

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

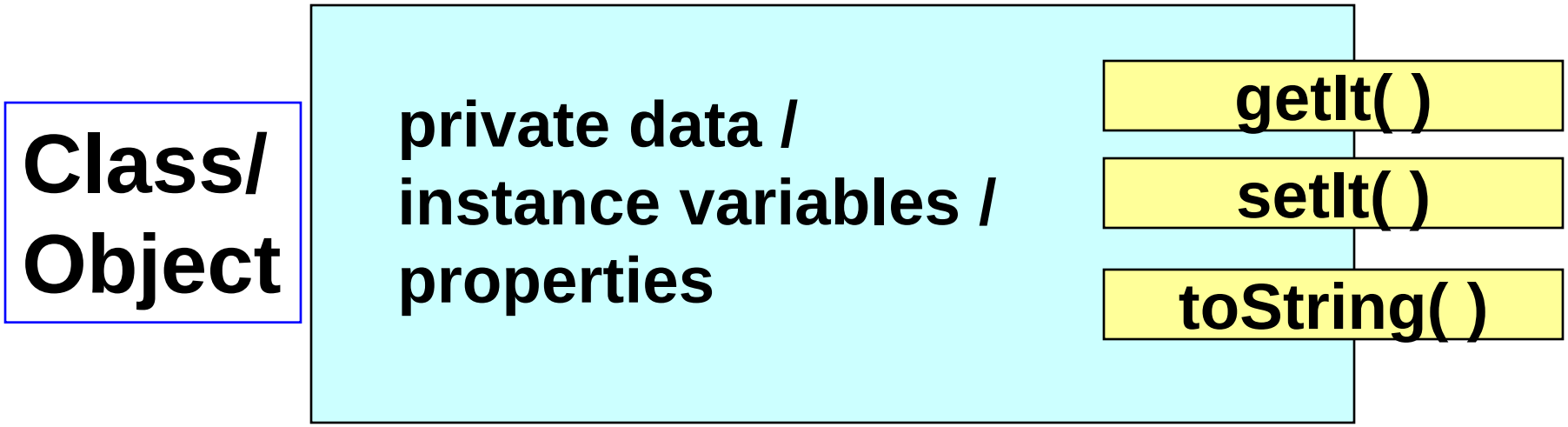
Accessor methods are methods that retrieve or grant access to the properties of an object, but do not make any changes.

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.

encapsulation

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

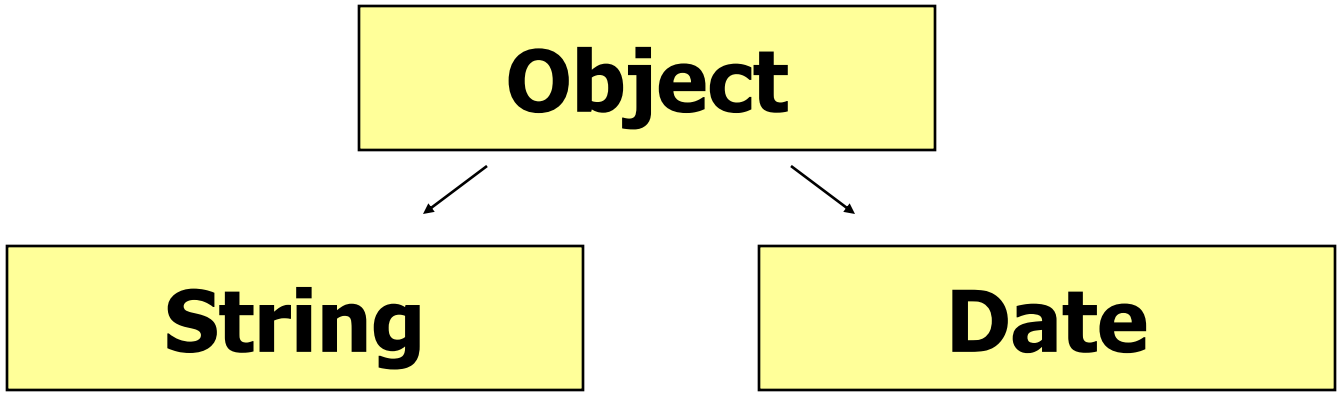


Open triangle.java triangleRunner.java

Object

class Object

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



All classes extend Object!

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

class Object

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

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

Open monsterone.java

Private/Public

What does public mean?

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

What does private mean?

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

Open private.java

Constructors

Constructors

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

Open monstertwo.java

equals

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));
```

equals()

```
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));
```

OUTPUT

false

Open equals.java

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

```
class Monster{
  private int height;                 //default assinged to 0
  private double weight;              //default assinged 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;
  }
}
```

Open overload.java

The Monster Class

Lab 18a

Monster Object Diagram

Monster

- Monster() - constructors
- setX( ) - modifiers
- getX( ) - accessors
- toString( ) - accessor

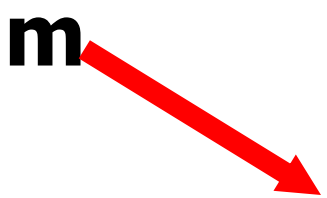
```
class Monster
{
  //instance vars / data fields

  public Monster(){
    //code
  }

  public void setX( params ){
    //code
  }

  public int getX(){
    //code
  }

  public String toString() {
    //code
  }
}
```

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
}
```

Monster Instantiation 1

```
Monster m = new Monster();
```



MONSTER

Properties

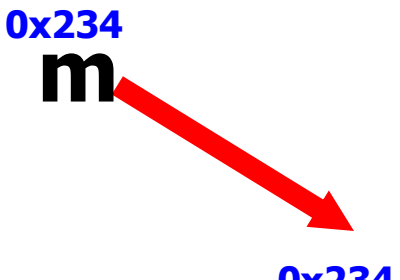
– height – 0 weight - 0 age - 0

methods

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

Monster Instantiation 2

Monster m = new Monster(23);

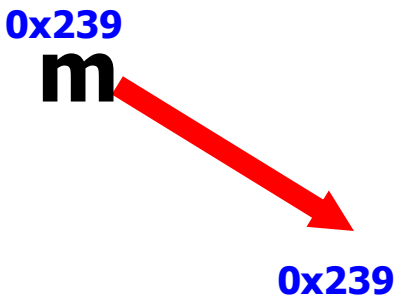


MONSTER
Properties
– height – 23 weight – 0 age - 0
methods

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

Monster Instantiation 3

Monster m = new Monster(23, 45);



MONSTER

Properties

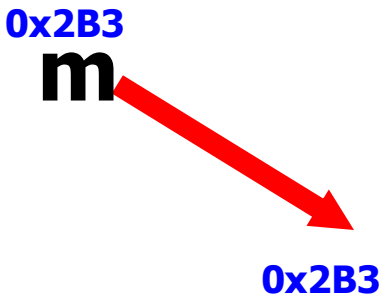
– height – 23 weight – 45 age - 0

methods

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

Monster Instantiation 4

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



MONSTER

Properties

– height – 23 weight – 45 age - 11

methods

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

Start work on Lab 18a