

CSE 8A Lecture 17

- Reading for next class: 11.3
- Today's goals:
 - Learn how to design and implement your own classes
- PSA 8 due Monday 3/4/13 11:59pm
 - Individual programming
 - PSA 8 Interview Due Thursday 3/7/13
- Interim Exam 4 Friday (3/8)

About Exam #4

- It will focus on **(Exam #3)** fundamentals, loops, decisions, methods, `copyPicture()`
- Lecture Slides:
 - **Lecture 11 (slides: 2-6, 14-17)**
 - **Lecture 12 (slides: 11-22)**
 - **Lecture 13 (slide: 2)**
 - **Lecture 14 (slides: 2-4)**
- Practice the following
 - **Writing code from scratch (PSA8 – understand what your code is doing)**
 - **Primitive data types:** boolean, char, int, double
 - **Literal values (int: 3 double: 3.14)**
 - **String, System.out.println(), System.out.print()**

Classes and objects

- **Classes** and **objects** are essential concepts in **object-oriented** programming languages like Java
- Classes and objects in programming languages are something like classes and objects in the real world
- Understanding the parallels can help you understand the concepts!

Objects

- Real world **objects** have:
 - **Properties**
 - For example, a car has a price, a color, an owner, a mileage rating, a location, a weight, an amount of gas in its tank...
 - **Behaviors**
 - For example, a car can be sold (change its owner), can move (change its location), can be filled (change the amount of gas in its tank)...
- Software objects have properties and behaviors too...

Classes

- Real world objects are instances of one or more **classes**
 - i.e. any car object is an instance of the class “Car”
 - An object’s class determines what properties it has and what behaviors it has
 - Note: a class may exist, without any objects (example: the class of unicorns!). The class comes first.
- Software objects are instances of classes too...

Chapter 11: Creating Classes

- An object gets the kinds of properties and behaviors it has from the definition of the class it is created as an instance of
- So, designing and defining a class is very important!
- Think: what properties and behaviors do you want your objects to have?

And define the class accordingly

Instance Variables and Methods

- Properties of an object are determined by its *instance variables*
 - What types are they? What values do they have?
 - (also known as ‘fields’ or ‘member variables’)
- Behaviors of an object are determined by its *instance methods*
 - What parameters do they take? What values do they return (if any)? What do they *do*?
 - *Constructors* are a special kind of method...

Constructors

- When an object is created, its instance variables need to be initialized
- This is the job of a constructor method
- Constructor methods in a class:
 - Always have the same name as the class
 - Never have a return type (not even void)
 - Can take parameters
 - Can be overloaded
 - Are only called when an object is instantiated

- 1) Solo: (60 sec)
2) Discuss/Group (2 min)

A **Point** class

The **Point** of Java: objects and classes

in a file named **Point.java**

```
public class Point
{
    private double x;
    private double y;

    public Point(double x_in, double y_in)
    {
        this.x = x_in;
        this.y = y_in;
    }

    public static void main( String[] args )
    {
        double d = 42.0;
        Point p;
        p = new Point(d, 42.0);

        Point q = new Point(p.x, 42)
    }
}
```

What are the *instance variables* in Point?

- A. x, y
- B. x_in, y_in
- C. x, y, x_in, y_in
- D. p, q
- E. x, y, p, q

- 1) Solo: (60 sec)
2) Discuss/Group
(2 min)

A Point class

The **Point** of Java: objects and classes

```
public class Point
{
    private double x;
    private double y;

    public Point(double x_in, double y_in)
    {
        this.x = x_in;
        this.y = y_in;
    }

    public static void main( String[] args )
    {
        double d = 42.0;
        Point p;
        p = new Point(d, 42.0);

        Point q = new Point(p.x, 42)
    }
}
```

How arguments (parameters) does the Point constructor take?

- A. 0
- B. 1
- C. 2
- D. 0 or 2 (you can call it with either)
- E. 0, 1 or 2

- 1) Solo: (60 sec)
- 2) Discuss/Group: (2 min)

A Point class

The **Point** of Java: objects and classes

```
public class Point
{
    private double x;
    private double y;

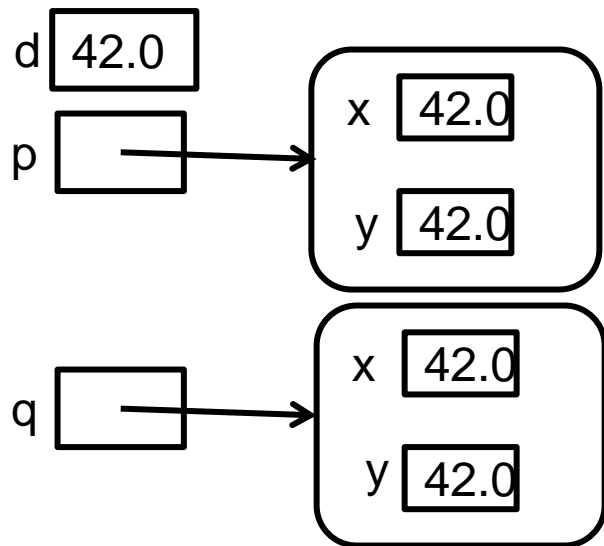
    public Point(double x_in, double y_in)
    {
        this.x = x_in;
        this.y = y_in;
    }

    public static void main( String[] args )
    {
        double d = 42.0;
        Point p;
        p = new Point(d, 42.0);

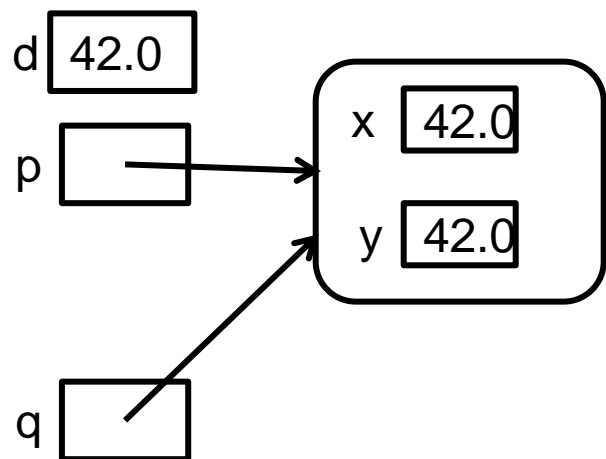
        Point q = new Point(p.x, 42)
    }
}
```

Draw a picture of the memory model at the end of main (vote on next slide)

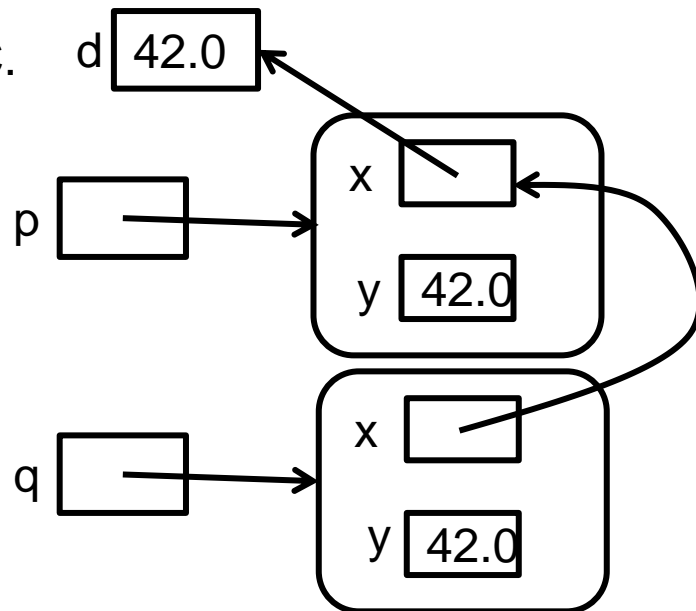
A.



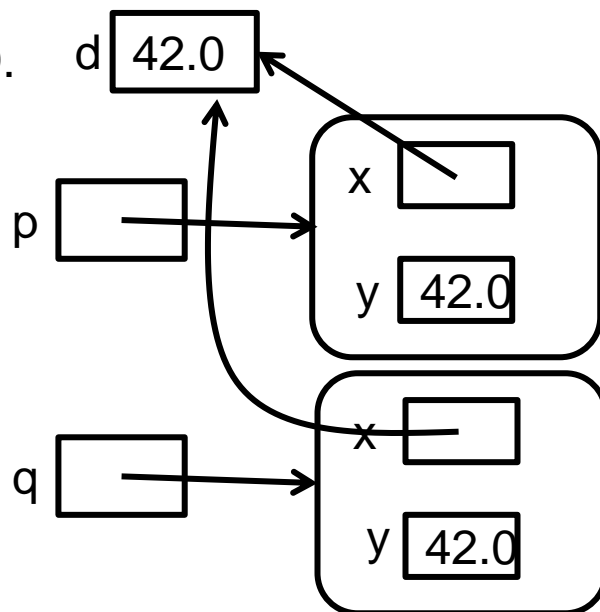
B.



C.



D.



- 1) Solo: (60 sec)
2) Discuss/Group:
(2 min)

A Point class

The **Point** of Java: objects and classes

```
public class Point
{
    private double x;
    private double y;

    public Point(double x_in, double y_in)
    {
        this.x = x_in;
        this.y = y_in;
    }

    public static void main( String[] args )
    {
        double d = 42.0;
        Point p;
        p = new Point(d, 42.0);

        Point q = new Point(p.x, 42)
        d = 65.0;
        p.x = 55.0;
        System.out.println( d + ", " + p.x + ", " + q.x )
    }
}
```

- What does the code print?
- A. 42.0, 42.0, 42.0
 - B. 65.0, 55.0, 65.0
 - C. 55.0, 55.0, 42.0
 - D. 65.0, 55.0, 55.0
 - E. 65.0, 55.0, 42.0

Another Example: the Pixel class

- Look at code in Pixel.java
- Identify: constructors, instance variables, instance methods

Class, Field, or Method?

- A class is a **type** of thing
- A field (instance variable) is a **property**, that might possibly have different values at different times
- A method is an **action** that can be performed
- So usually class and field names are nouns, method names are verbs...

Another example of a class

- Class: **Species**
- Instance Variables:
 - **name**, a String
 - **population** on 7 continents, an array of 7 ints
 - **growthRate**, a double
- Constructor: one that takes no arguments, and:
 - initializes **name** to “No Name Yet”
 - initializes **population** to all 0
 - Initializes **growthRate** to 33.3

- 1) Solo: (60 sec)
2) Discuss/Group:
(2 min)

How many errors are there in this code (and what are they)

```
public class Species

    private String name;
{

    public static void main(String[] args)
    {
        double[] population;
        double growthRate;
    }

    public Species()
    {
        String name = "No Name Yet";
        double[] population = {0,0,0,0,0,0,0};
        growthRate = 33.3;
    }

}
```

A. 2

B. 3

C. 4

D. 5

E. ≥ 6

Lab 7 quiz

1) In this example, what **number** should you divide totred by to compute the average red value? Write the **number** in the blank.
(Assume no IndexOutOfBoundsException is thrown when this code runs.)

```
int i, avgred, totred = 0;
```

```
for( i = x-3 ; i<x+3 ; i++ )
```

```
{
```

```
    totred = totred + this.getPixel(i,y).getRed();
```

```
}
```

```
avgred = totred / 6 ;
```

Lab 7 quiz

Write the equivalent statement using the other variable for each statement

(A and B) below. In A) write the statement using foo,

in B) write the equivalent statement using noise.

Assume the following declarations:

```
Sound noise = new Sound(FileChooser.pickAFile());
```

```
SoundSample[] foo = noise.getSamples();
```

// Write the equivalent using foo and not noise

```
noise.getSampleValueAt(i);
```

```
foo[i].getValue();
```

// Write the equivalent using noise and not foo

```
foo[30].setValue(3210);
```

```
noise.setSampleValueAt(30, 3210);
```

TODO

- Prepare for Exam #4
- Finish PSA 8 by Monday 11:59pm (~midnight)