CSE 8A Lecture 18

- Reading for next class: 11.4-11.5
- Today's goals:
  - More practice with designing classes
  - Tracing code and creating memory models
- PSA 9 (classes) due next Monday (3/11)
  - Individual (no partner)
- PSA8 due tonight (interview by Thursday)

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

A **Point** class

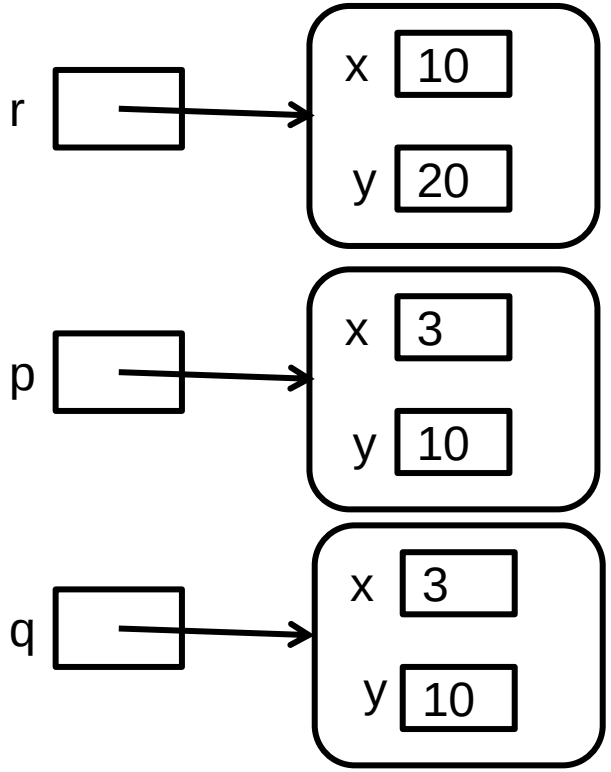
The **Point** of Java:

objects and classes

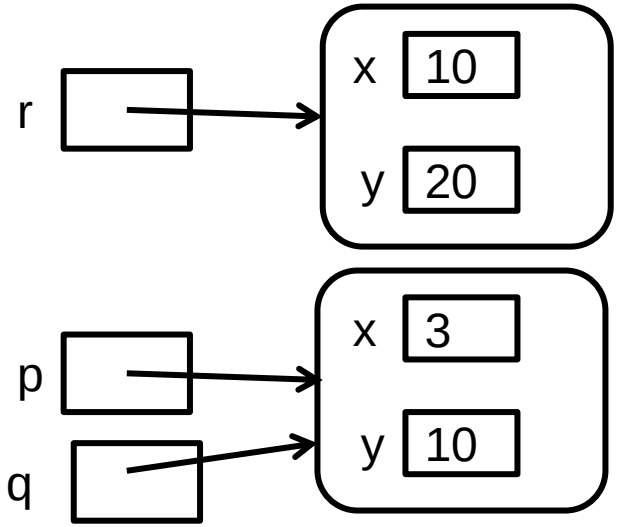
```
public class Point
{
  private int x;
  private int y;

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

  public static void main( String[] args )
  {
    Point r = new Point(10, 20);
    Point p = new Point(3, r.x);
    Point q = p;



}
```

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

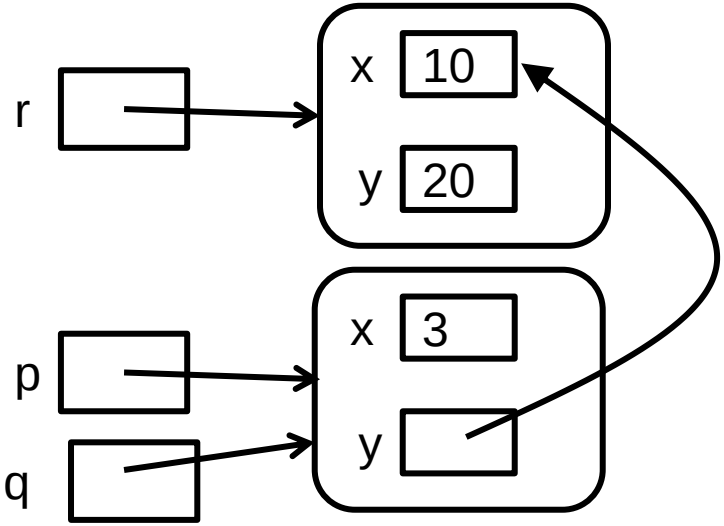
A.



B.



C.



D. None of these

E. I don’t know

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

A **Point** class

The **Point** of Java:

objects and classes

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

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

  public static void main(String[] args)
  {
    Point r = new Point(10, 20);
    Point p = new Point(3, r.x);
    Point q = p;
    r.x = 1;
    q.y = r.x + p.x;

}
```

What are the values of r, p, and q when this code completes?

	r	p	q
A.	(1, 20)	(3, 1)	(3, 4)
B.	(1, 20)	(1, 1)	(3, 2)
C.	(10, 20)	(3, 13)	(3, 13)
D.	(1, 20)	(1, 13)	(1, 13)

E. None of these

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

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

```
public class Species
{
  private String name;
}

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

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

}
```

All instance variables and methods have to go inside the class { }

All instance variables have to go outside of method definitions

Instance variables should be private

Refer to instance variables in a constructor; do not redeclare them

Visibility of Instance Variables

- Class design rule of thumb: *make all instance variables* ***private***
  - *"**private**"* means: visible only inside this class
  - So a private instance variable or instance method cannot be seen from outside the class
  - Making an instance variable private prevents incorrectly setting its value by malicious or careless users of the class

Private instance variables in Species

```
public class Species
{

  ///////// fields ////////////
  private String name;
  private int[] population;
  private double growthRate;

 /////// constructors ///////////
  public Species()
  {
    name         = "No Name Yet";
    population   = {0,0,0,0,0,0,0};
    growthRate   = 33.3;
  }


 /////// methods ////////////////
}
```

Getter and Setter methods

- Q: Instance variables correspond to properties of an object… if they are private and hidden inside, how can they interact with other objects?
- A: Define public instance methods which give controlled, safe access to the private instance variables
  - If the method can change an instance variable, it is a "mutator" or "modifier" or "setter" method
  - If it only returns the value of an instance variable, it is an "accessor" or "getter" method

1) Solo: (45 sec)
2) Discuss/Group: (2 min)

Which of following would you select for "getter" method signatures for Species class?

```
public void getName();
public void getPopulation();
public void getGrowthRate();
```

```
public String getName();
public int[]  getPopulation();
public double getGrowthRate();
```

```
public void getName(String newName);
public void getPopulation(int newPop);
public void getGrowthRate(double newGrowthRate);
```

```
private String getName();
private int[]  getPopulation();
private double getGrowthRate();
```

1) Solo: (45 sec)
2) Discuss/Group: (2 min)

Which of following would you select for "setter" method declarations for Species class?

```
public void setName();
public void setPopulation();
public void setGrowthRate();
```

```
public String setName();
public int[]  setPopulation();
public double setGrowthRate();
```

```
public void setName(String newName);
public void setPopulation(int[] newPop);
public void setGrowthRate(double newGrowthRate);
```

```
public void setName(String newName);
public boolean setPopulation(int[] newPop);
public void setGrowthRate(double newGrowthRate);
```

1) Solo: (30 sec)
2) Discuss/Group: (2 min)

Return type for Setters

- A getter method should have a non-void return type
- A setter can be designed in several ways:
  - void: just change the values of the instance variable(s), don't return anything
  - boolean: return true if the setting was successful and false if not (for example if setting would be 'illegal')
  - The type of the value that is being changed: return the previous value

1) Solo: (45 sec)
2) Discuss/Group: (2 min)

Overloading: Which are legal overloads?

```
public Species()
public Species(String newName);
```

```
public boolean setGrowthRate(double gr)
public void    setGrowthRate(double gr)
```

```
public void setPopulation(int northAmerica,
                          int southAmerica,
                          int europe,
                          int asia,
                          int africa,
                          int australia,
                          int antarctica)
public void setPopulation(int[] a)
```

A. 1

B. 2

C. 3

D. 1 and 3

E. 1 and 2

1) Solo: (45 sec)
2) Discuss/Group: (2 min)

Terminology Check

1. Declaration
2. Instantiation
3. Initialization

```
foo = new double[5];
```

```
for(int i = 0 ; i < foo.length ; i++)
{ foo[i] = -11.5; }
```

```
double [] foo;
```

Draw a memory model for this code:

```
public Species( String newName, int[] newPop, double newGR )
{
  name       = newName;
  population = new int[newPop.length];

  for( int i=0 ; i< this.population.length ; i++ )
    population[i] = newPop[i];

  growthRate = newGR;
}
```

TODO

- Reading for next class: 11.4-11.5
- Submitted PSA8 (if you haven’t done it already)
- Start PSA9
  - Do this one individually (no team programming)