ITEC1620
Object-Based Programming

Lecture 12

Methods, APIs, and Object Diagrams

Methods

- Methods are functions that allow interaction with the data in an object
- Getting object information
  - int input = userInput.nextInt();
  - double x = point.getX();
- Some methods allow setting object information

Methods II

- Methods are like mathematical functions

- What happens with a = f (x, y)?
    - largest = max(x, y);

The Pieces of a Mathematical Function

- name → max
- parameters → x, y
- output/result → larger value

Functions to Methods

● name	→	identifier
● parameters	→	parameters (with datatypes)
● output/result	→	return value

Signature for a Method

<access modifiers> <return datatype> <identifier> (<parameters>)

Example

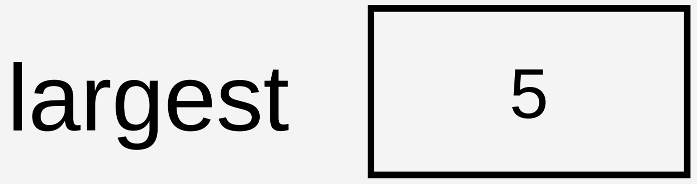
In the Math package

```
static int max ( int x, int y )
```

Returns the greater of two int values

Example II

```
int largest = Math.max (3, 5);
```

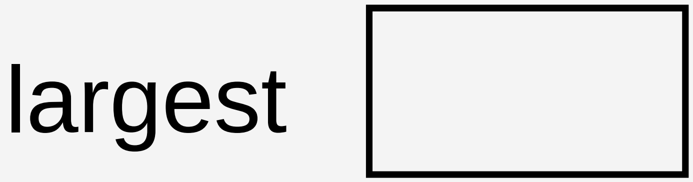


largest 5

Methods in Action

```
int largest = Math.max (3, 5);
```

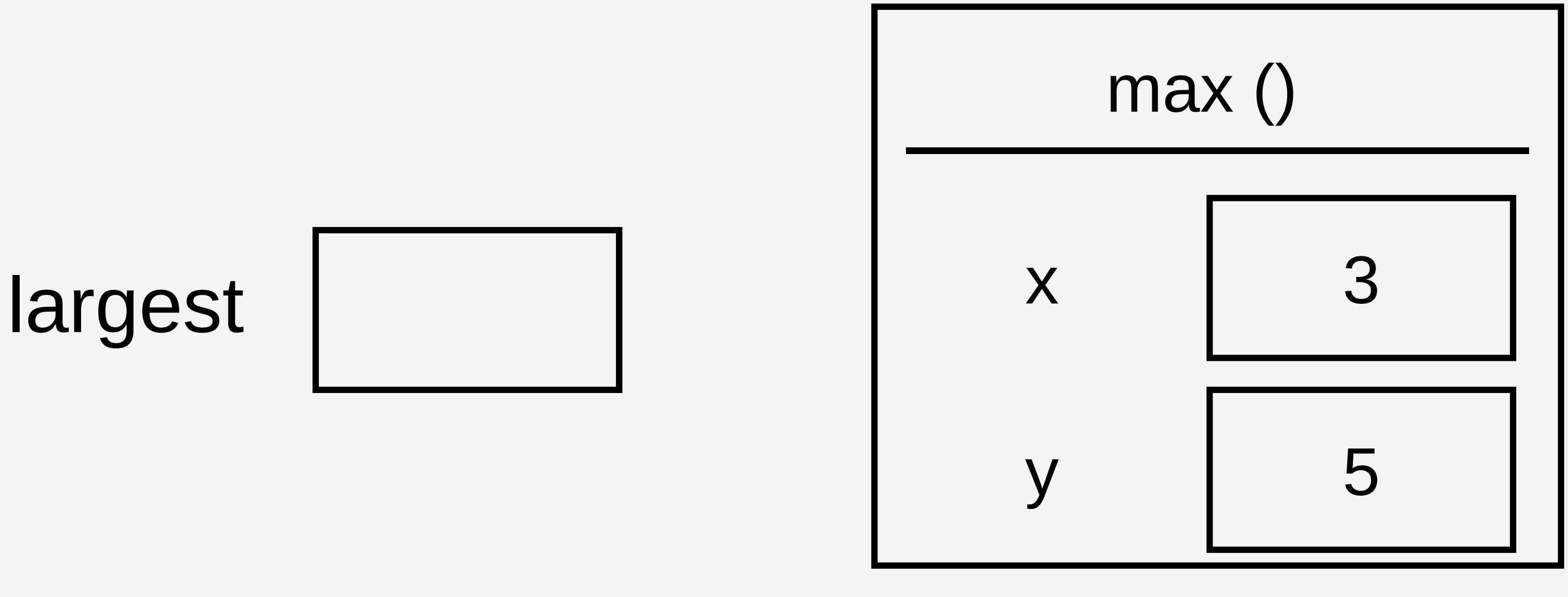
- Declare identifier largest, reserve space



largest

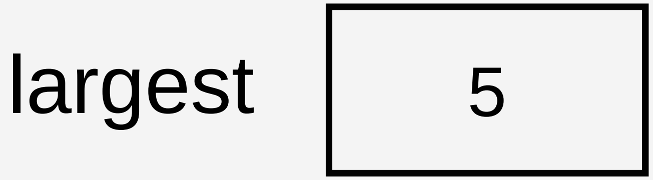
Methods in Action II

- Find method max with parameters (int, int)
- Copy parameters into method



Methods in Action III

- Execute method
- Return value back to calling statement (if return type is not void)



largest

5

Questions?

Using Classes

- A “class” is a template for objects
- The Point class specifies that each Point object will have two ints – an x and a y coordinate

```
pointA.x
pointA.y
```

Using Classes II

- A class is a user-defined datatype
- Like primitive datatypes, Java allows an unlimited number of instances of user-defined datatypes

```
    Point pointA;
    Point pointB;
```

- An instance of a class is an object

Instantiating Objects

- Java does not know how to initialize your datatype
- You must use a constructor
  - A constructor is a special method that has the same name as the class
- The constructor allocates space in memory for all the fields of the class
  - Space allocation requires “new”

Instantiating Objects II

- You must use the constructor before using your objects

```
Point pointA;
pointA.x = 0;              // error

Point pointB = new Point();
pointB.x = 0;              // OK
```

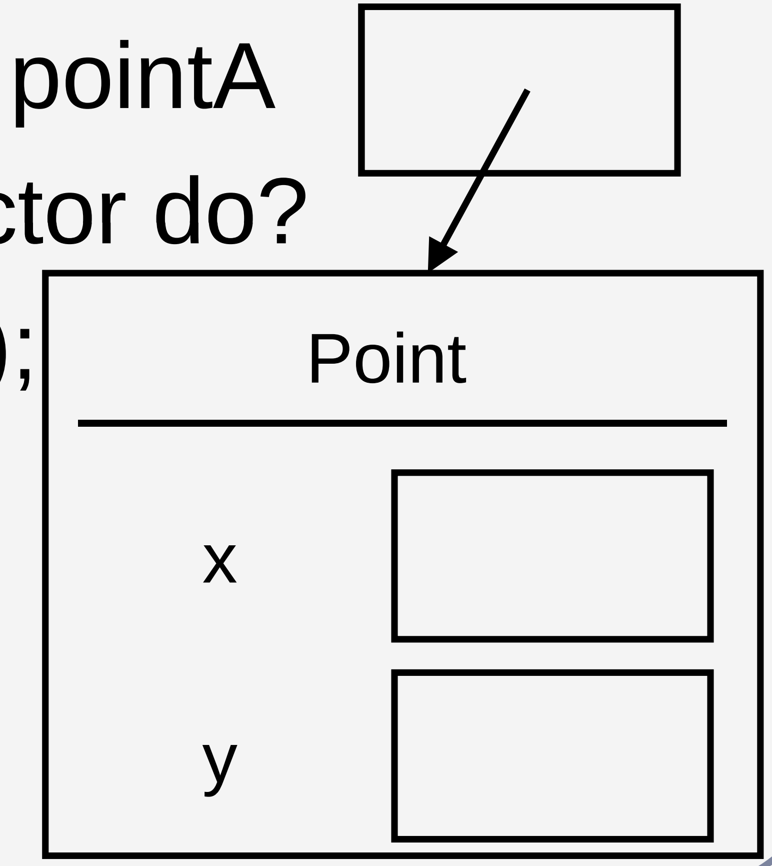
Instantiating Objects III

- What does the declaration do?

```
Point pointA;
```

- What does the constructor do?

```
pointA = new Point();
```

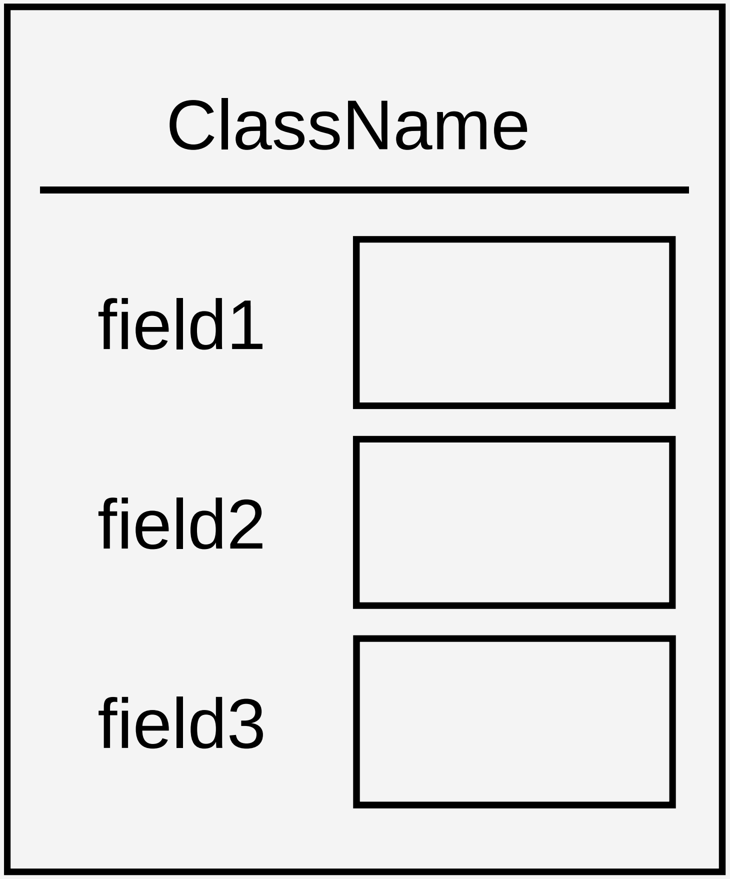


Object Diagrams

- Calling a constructor creates an object – an instance of a class
- Each instance of a class has its own copy of each field
  - Each point has an x coordinate and a y coordinate
- We draw these fields in an “object diagram”

Object Diagrams II

- Object diagrams have:
  - A big box
  - The class name
  - Little boxes for each field



References

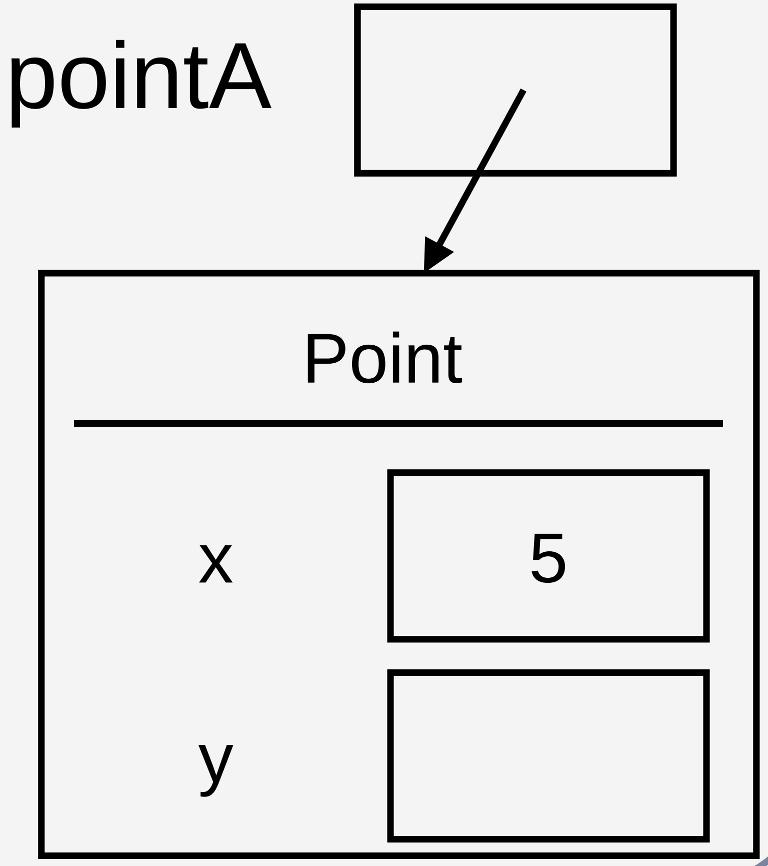
- Classes can be compiled separately
- The compiler doesn’t know how big each (user-defined) class will be
- How much space to reserve?
  - Reserve space for a reference
- A reference is the memory location of an object
- Use “arrows” in class diagrams

Dot operator

- The identifier memory location has a reference
- How do you access the fields of the object?
- Use the “dot” operator
- The dot means “follow the arrow”

Dot operator II

```
Point pointA = new Point();
pointA.x = 5;
```

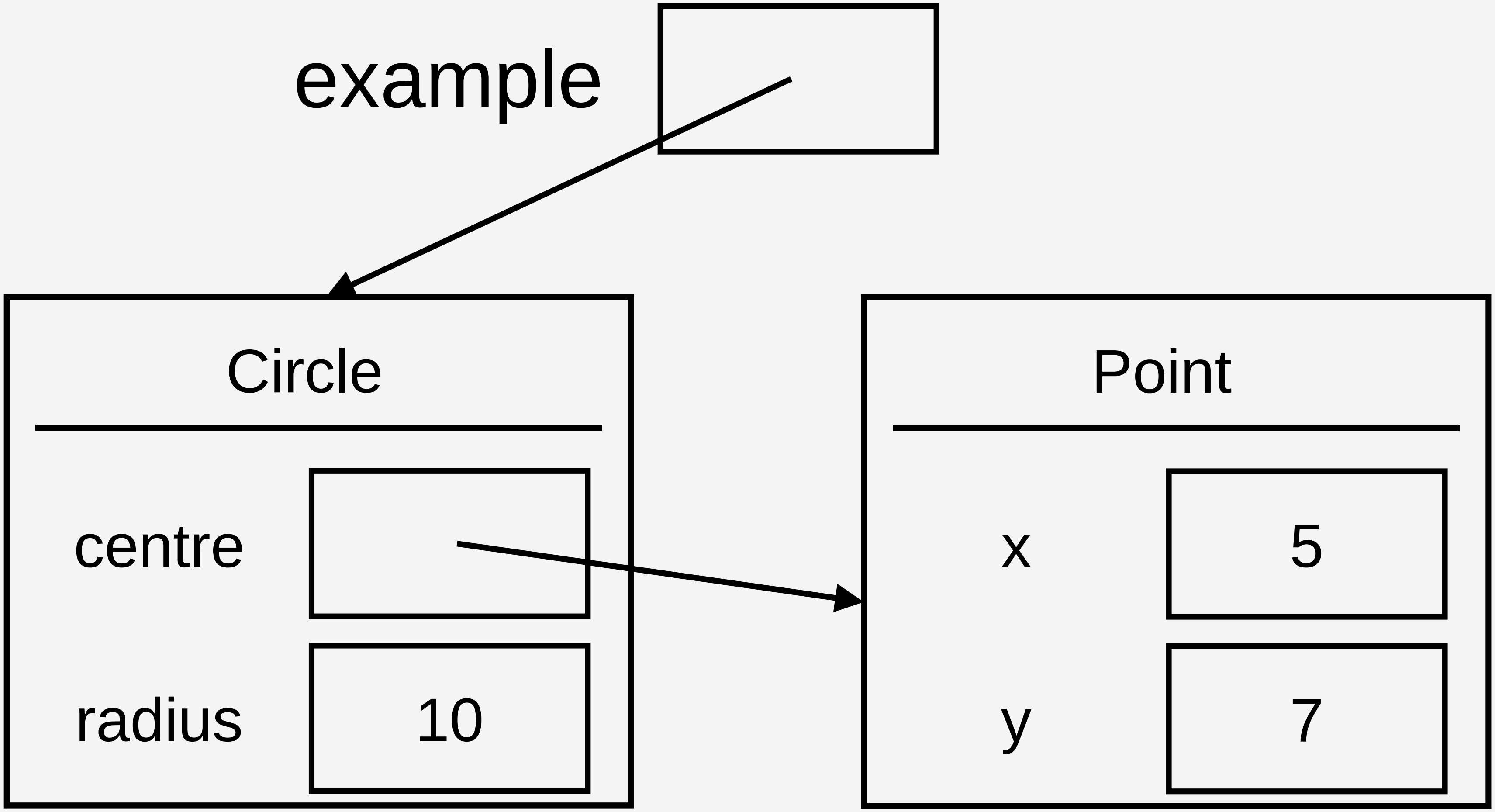


Questions?

Example Object Diagram

- Draw the object diagram(s) for a Circle of radius 10 centred at 5,7

```
Circle example = new Circle();
example.centre = new Point();
example.centre.x = 5;
example.centre.y = 7;
example.radius = 10;
```



example

Circle

centre

radius 10

Point

x 5

y 7

Example Object Diagram II

- Change the centre Point to a Point at the origin

```
Point origin = new Point();
origin.x = 0;
origin.y = 0;
example.centre = origin;
```

Example Program with Objects

- Write a code fragment that determines if two Points can be the diagonal corners of a square

```
// Point upperLeft, lowerRight;
```

```
int verticalDist = upperleft.y - lowerRight.y;
int horizontalDist = lowerRight.x - upperleft.x;
if (verticalDist == horizontalDist)
    System.out.println(“can be a square”);
else
    System.out.println(“cannot be a square”);
```

Readings and Assignments

- Text sections (5th, 6th, or 7th edition)
  - 3.1, 4.1