Today’s big ideas

• Two kinds of data in Java: **primitives** and **objects**

• We refer to an object using a **reference**

• There is a difference between passing objects and primitives to a method
CS 2230
CS II: Data structures

Meeting 3: Objects and references
Brandon Myers
University of Iowa
An example from the doctor

• We need to build a system to track patients

• First, patients have a name and height (in inches)

```java
class Patient {
    String name;
    int height;
}
```

defines the class Patient
Creating a patient

class Patient {
    String name;
    int height;

    // constructor (says how to initialize a new Patient)
    Patient(String n, int h) {
        name = n;
        height = h;
    }

    public static void main(String[] args) {
        // create a Patient
        Patient p1 = new Patient("Jane Doe", 65);
    }
}
**Updating a patient**

When patients come in for a check up, we want to update their height with the latest measurement.

```java
class Patient {
    String name;
    int height;

    Patient(String n, int h) {
        name = n;
        height = h;
    }

    void updateHeight(int newHeight) {
        height = newHeight;
    }

    public static void main(String[] args) {
        Patient p1 = new Patient("Jane Doe", 65);
        p1.updateHeight(70);  // call the method on p1
    }
}
```
Peer instruction

```java
public class MathStuff {
    static void squareIt(int x) {
        x = x*x;
    }

    public static void main(String[] args) {
        int a = 10;
        squareIt(a);
        System.out.println(a);
    }
}
```

What does the program print to the console?

a) a
b) 10
c) 100
d)
e) 10*10
public class MathStuff {
    static void squareIt(int[] x) {
        x[0] = x[0] * x[0];
    }

    public static void main(String[] args) {
        int[] a = new int[3];
        a[0] = 10;
        squareIt(a);
        System.out.println(a[0]);
    }
}

What does the program print to the console?

a) a
b) 10
c) 100
d) 
e) x[0]*x[0]
let’s work through that last example with diagrams

public class MathStuff {
    static void squareIt(int[] x) {
        x[0] = x[0] * x[0];
    }
}

public static void main(String[] args) {

    int[] a;
    a = new int[3];
    a[0] = 10;
    squareIt(a);
    System.out.println(a[0]);
}

References
public class MathStuff {
    static void squareIt(int[] x) {
        x[0] = x[0] * x[0];
    }

    public static void main(String[] args) {
        int[] a;
        a = new int[3];
        a[0] = 10;
        squareIt(a);
        System.out.println(a[0]);
    }
}

References
public class MathStuff {
    static void squareIt(int[] x) {
        x[0] = x[0] * x[0];
    }

    public static void main(String[] args) {
        int[] a;
        a = new int[3];
        a[0] = 10;
        squareIt(a);
        System.out.println(a[0]);
    }
}
public class MathStuff {
    static void squareIt(int[] x) {
        x[0] = x[0] * x[0];
    }

    public static void main(String[] args) {
        int[] a;
        a = new int[3];
        a[0] = 10;
        > squareIt(a);
        System.out.println(a[0]);
    }
}
public class MathStuff {
    static void squareIt(int[] x) {
        x[0] = x[0] * x[0];
    }

    public static void main(String[] args) {
        int[] a;
        a = new int[3];
        a[0] = 10;
        squareIt(a);
        System.out.println(a[0]);
    }
}
public class MathStuff {
    static void squareIt(int[] x) {
        x[0] = x[0] * x[0];
    }

    public static void main(String[] args) {
        int[] a;
        a = new int[3];
        a[0] = 10;
        squareIt(a);
        System.out.println(a[0]);
    }
}
Today’s big ideas

• Two kinds of data in Java: **primitives** and **objects**

• We refer to an object using a **reference**

• There is a difference between passing objects and primitives to a method