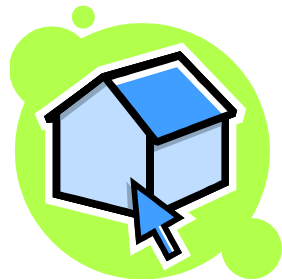


References

Array of References



What is a
What is a
reference?

References

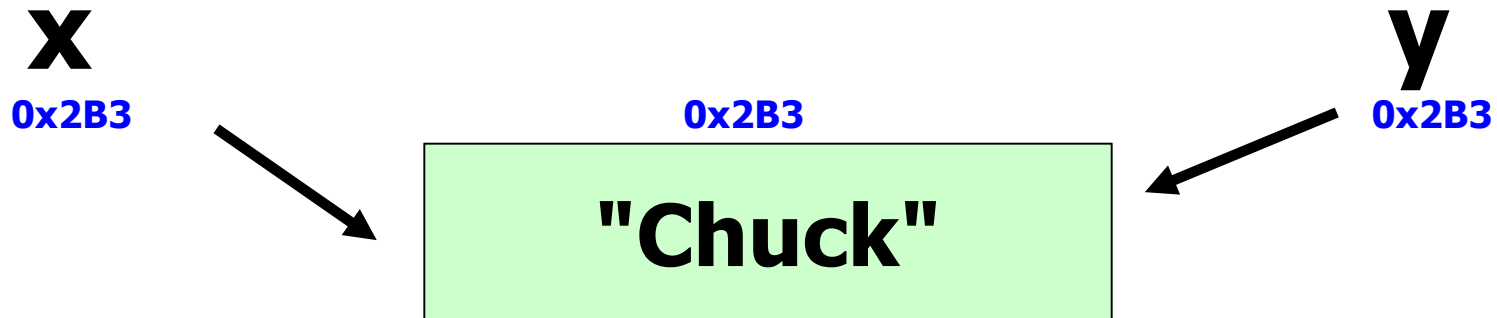
In Java, any variable that refers to an Object is a reference variable.

The variable stores the memory address of the actual Object.

References

```
String x = new String("Chuck");  
String y = x;
```

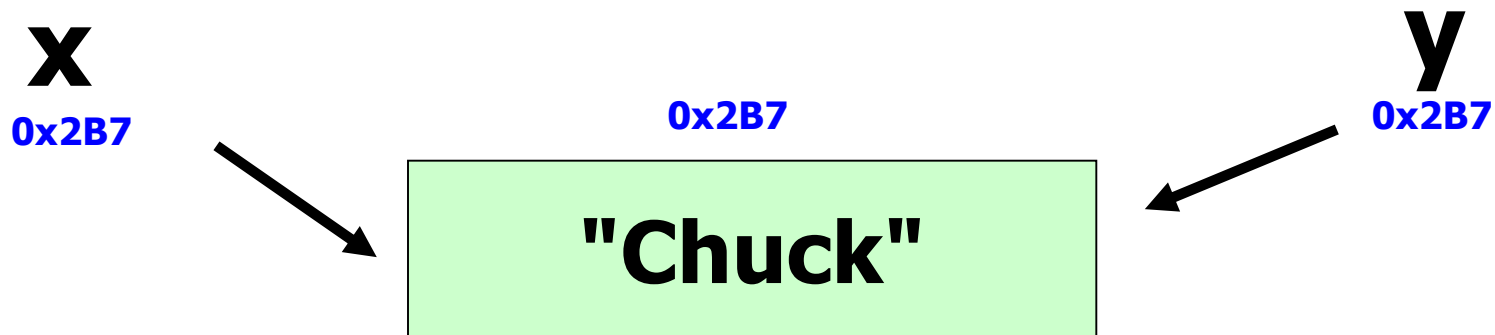
x and y store the same memory address.



References

```
String x = "Chuck";  
String y = "Chuck";
```

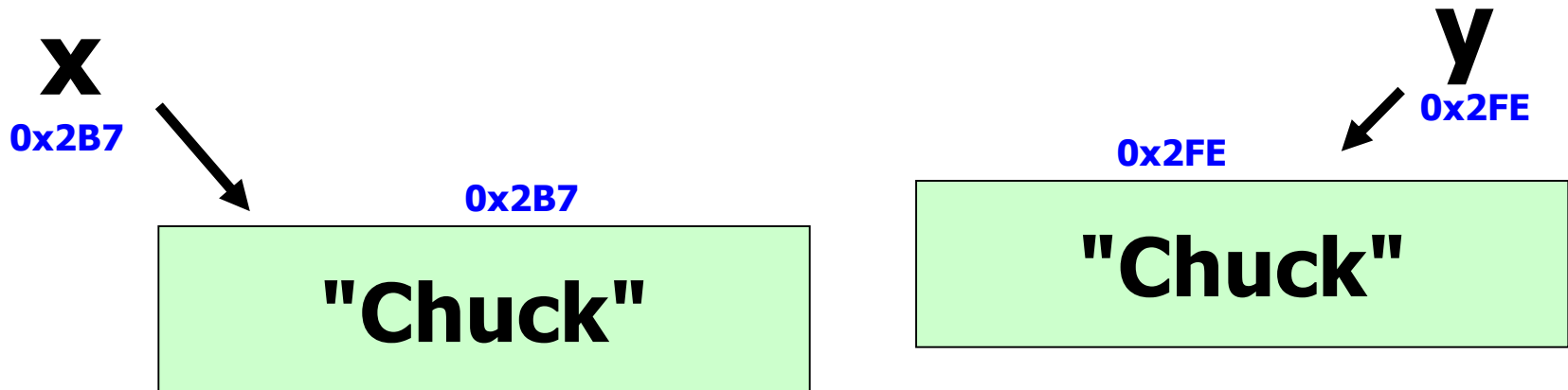
x and y store the same memory address.



References

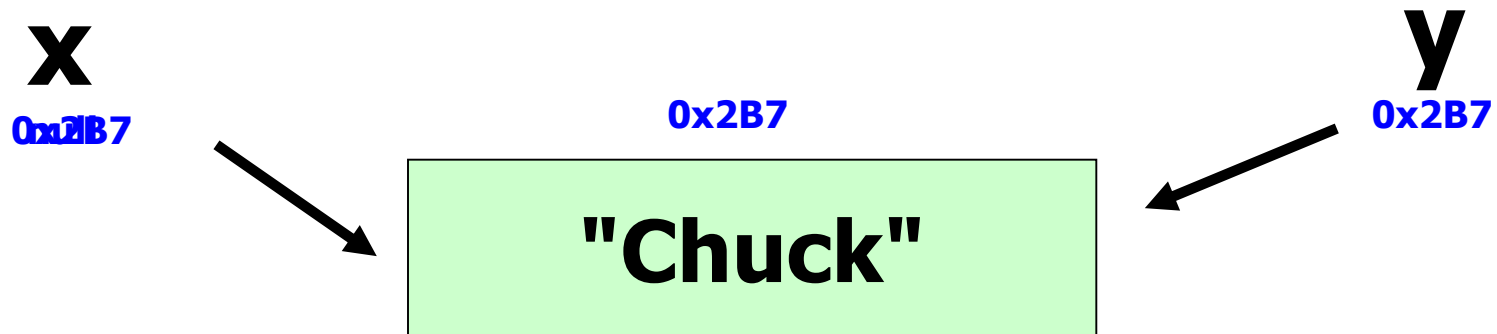
```
String x = new String("Chuck");  
String y = new String("Chuck");
```

x and y store different memory addresses.



References

```
String x = "Chuck";  
String y = "Chuck";  
x = null;
```



open references.java

Array of References

Array of References

```
String[] list = new String[50];  
//all 50 spots are null
```

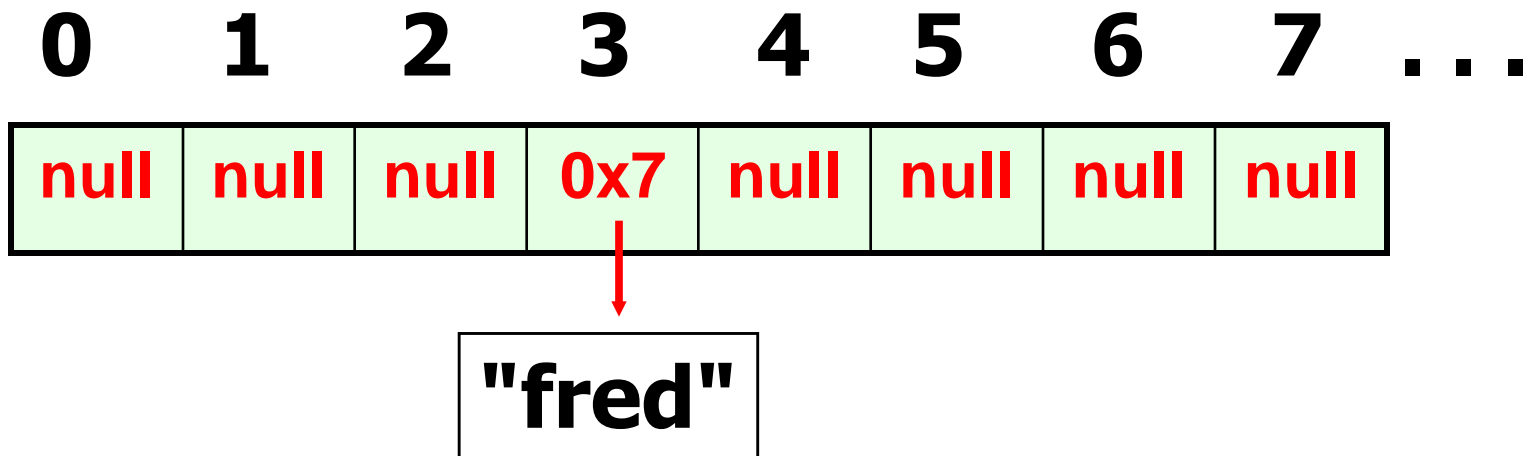
0 1 2 3 4 5 6 7 ...

null	null	null	null	null	null	null	null
------	------	------	------	------	------	------	------



Array of References

```
list[3] = "fred";
```



Open

arrayofreferencesone.java

Array of Monster References

class Monster

```
public 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)
```

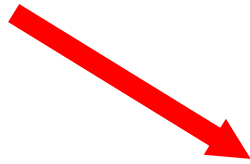
```
    { code }
```

```
}
```

Monster Instantiation 1

Monster m = new Monster();

m



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);

0x234

m

0x234

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);

0x239

m

0x239

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);

0x2B3

m

0x2B3

MONSTER

Properties

– height – 23 weight – 45 age - 11

methods

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

Array of References

```
Monster[] list = new Monster[5];
```

```
out.println(list[0]);  
out.println(list[1]);  
out.println(list[2]);  
out.println(list[3]);  
out.println(list[4]);
```

OUTPUT

null

null

null

null

null

Array of References

```
Monster[] list = new Monster[5];  
list[0] = new Monster();  
list[1] = new Monster(33);  
list[2] = new Monster(3,4,5);
```

```
out.println(list[0]);  
out.println(list[1]);  
out.println(list[2]);  
out.println(list[3]);
```

OUTPUT

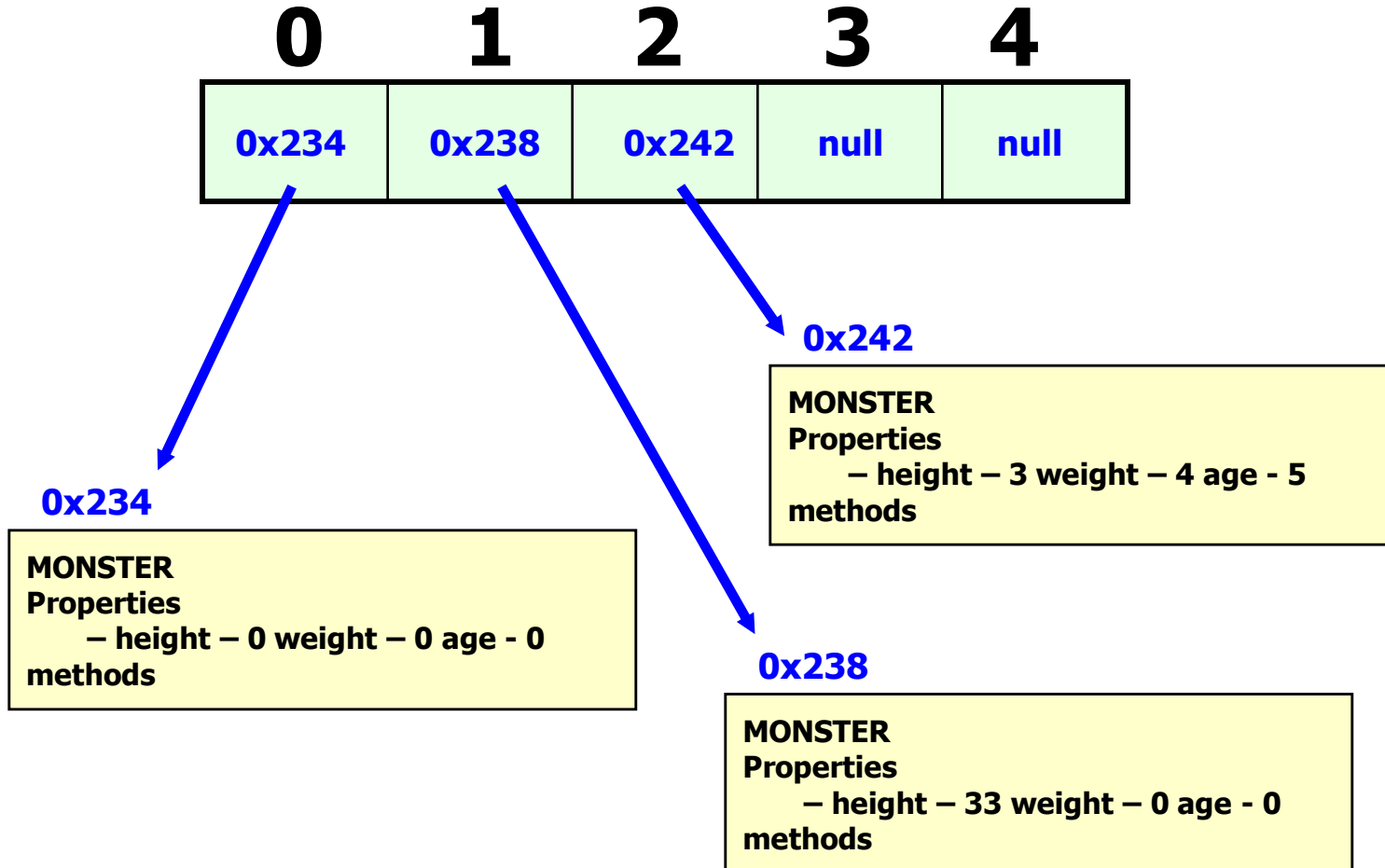
0 0 0

33 0 0

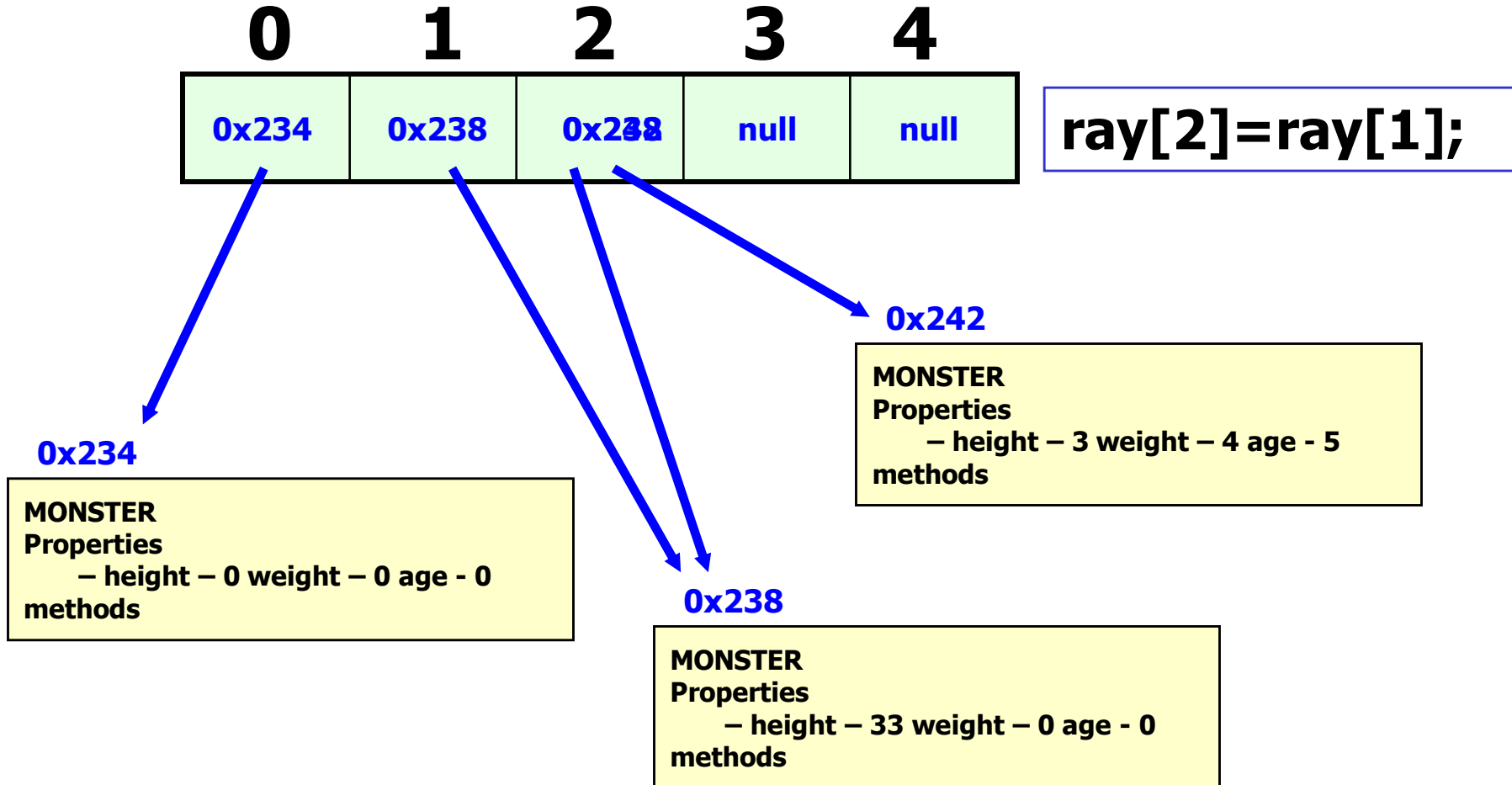
3 4 5

null

Array of References



Array of References



Open

arrayofreferencetwo.java

Array of References

```
public class Creature implements Comparable  
{  
    //data and constructors now shown  
  
    public void setSize(int girth){  
        size=girth;  
    }  
  
    //toString not shown  
}
```


Array of References

```
Creature[] creatures = new Creature[3];  
creatures[0]=new Creature(4);  
creatures[1]=new Creature(9);  
creatures[2]=new Creature(1);
```

```
out.println(creatures[0]);  
creatures[0].setSize(7);
```

```
out.println(creatures[0]);  
out.println(creatures[2]);
```

OUTPUT

4

7

1

Array of References

```
creatures[0].setSize(7);
```

0x242

**What
does this
store?**

**What
does the
. dot do?**

0x242

Creature

**The . dot grants access to the
Object at the stored address.**

Open

arrayofreferencesthree.java

Open

creatures.java

creaturesrunner.java

**Start work
on Lab 19a**