References

Array of References

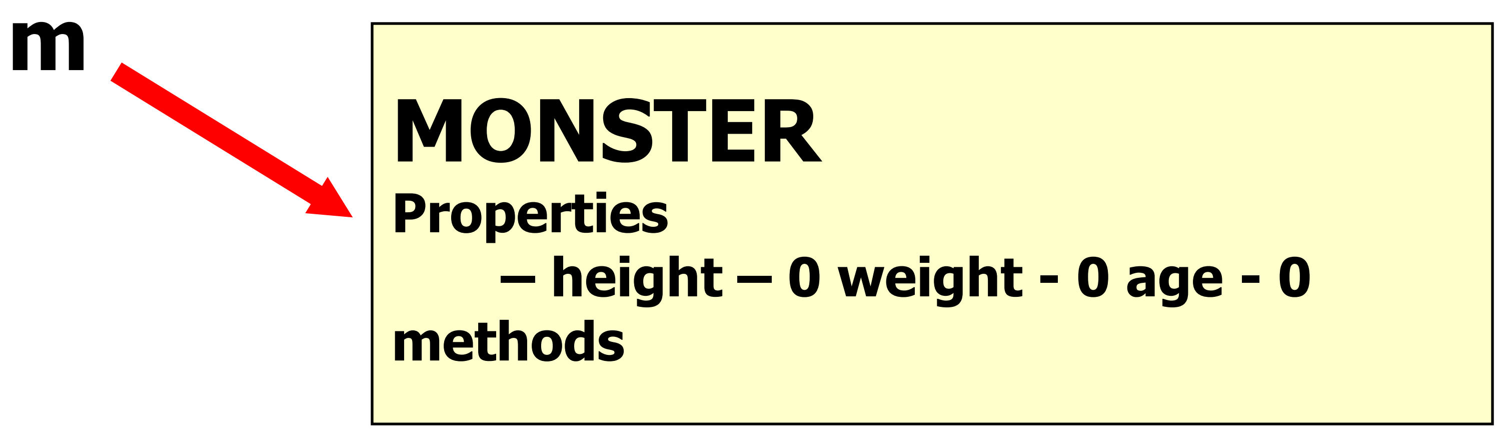
Object Instantiation

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

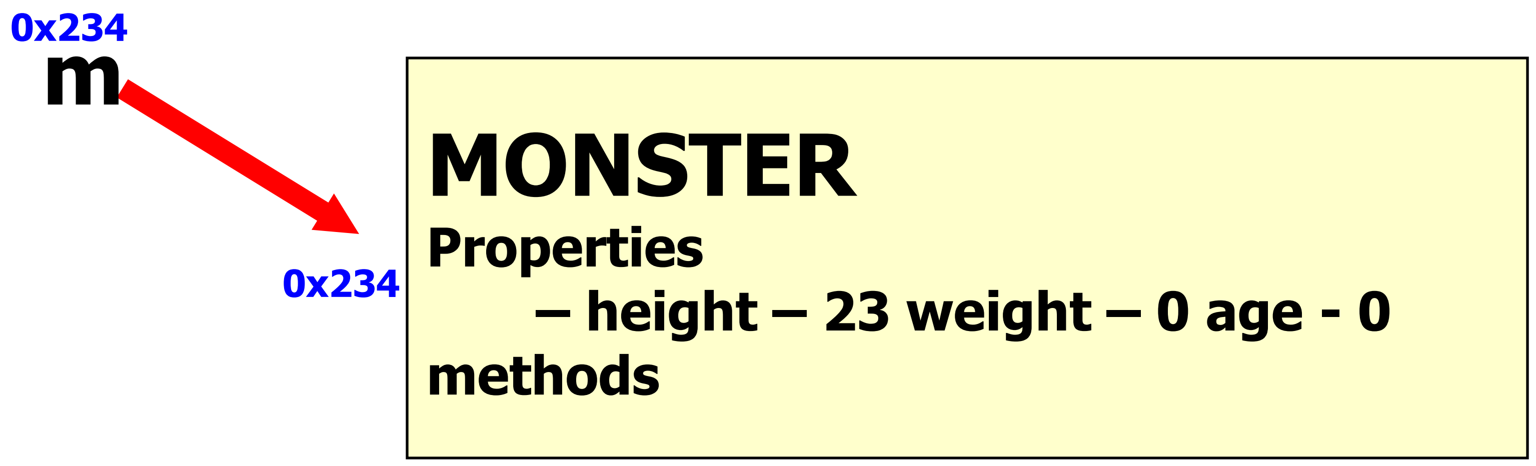
Monster m = new Monster();



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

Monster Instantiation 2

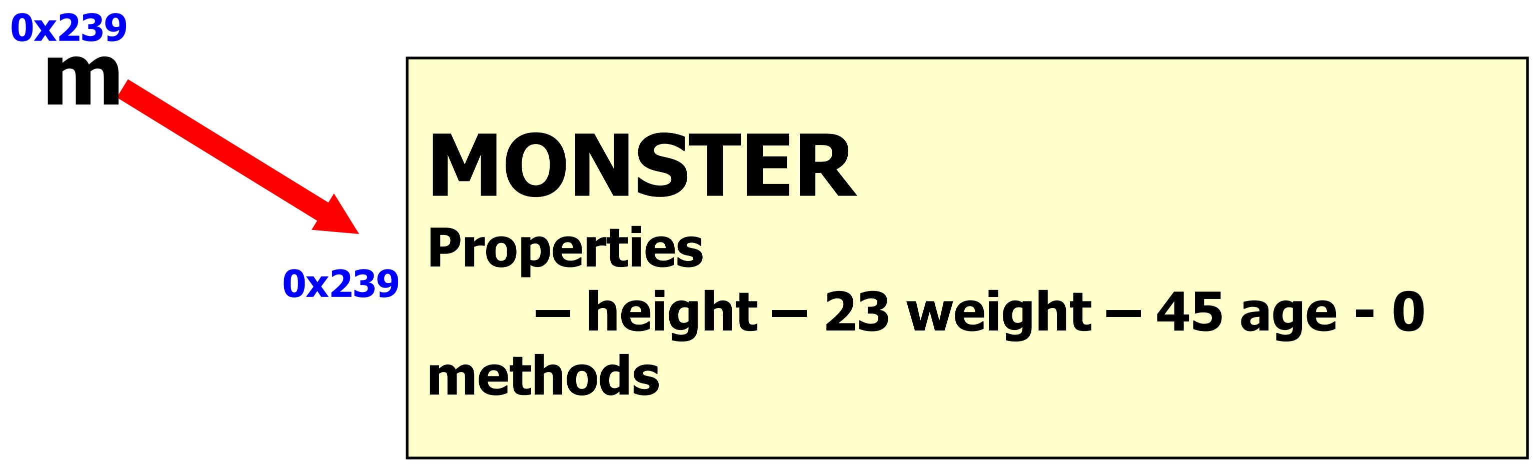
```
Monster m = new Monster(23);
```



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

Monster Instantiation 3

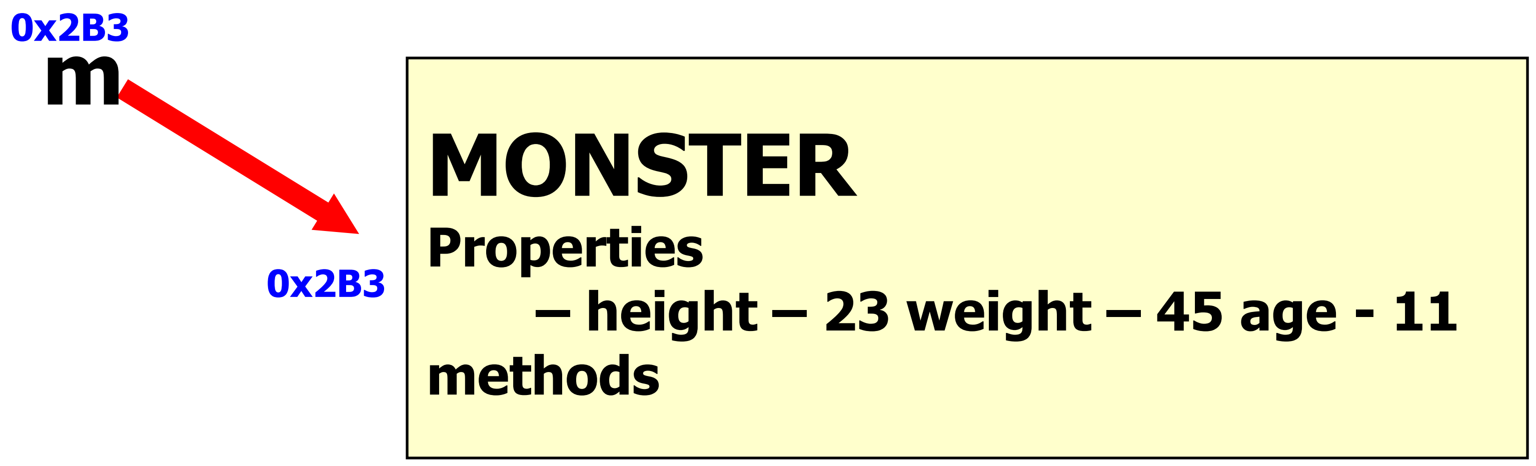
Monster m = new Monster(23, 45);



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

Monster Instantiation 4

```
Monster m = new Monster(23, 45, 11);
```



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

References

References

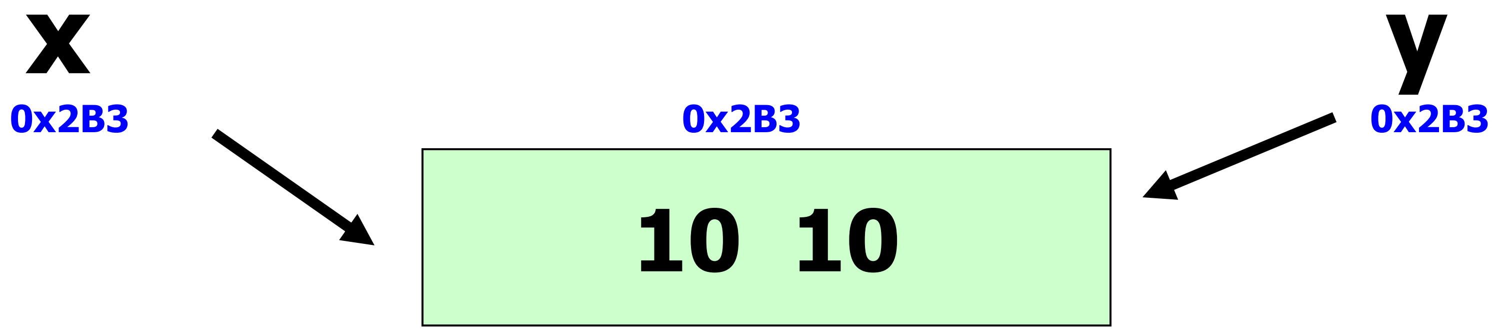
In Java, any variable that refers to an Object is a reference variable.

The variable stores the memory address of the actual Object.

References

```
Monster x = new Monster( 10, 10 );
Monster y = x;
```

x and y store the same memory address.



References

```
Monster x = new Monster( 10, 10 );
Monster y = x;

System.out.println( x == y );
System.out.println( x.equals( y )  );
```

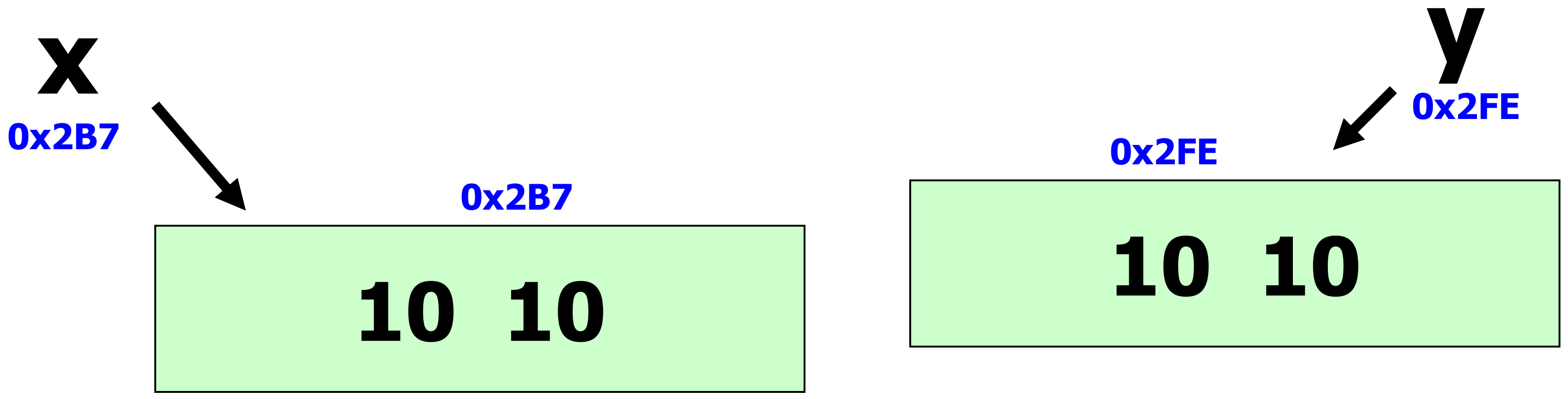
OUTPUT

```
true
true
```

References

```
Monster x = new Monster( 10, 10 );
Monster y = new Monster( 10, 10 );
```

x and y store different addresses.



References

```
Monster x = new Monster( 10, 10 );
Monster y = new Monster( 10, 10 );

System.out.println( x == y );
System.out.println( x.equals( y )  );
```

OUTPUT

```
false
true
```

open
references.java

Array of Monster References

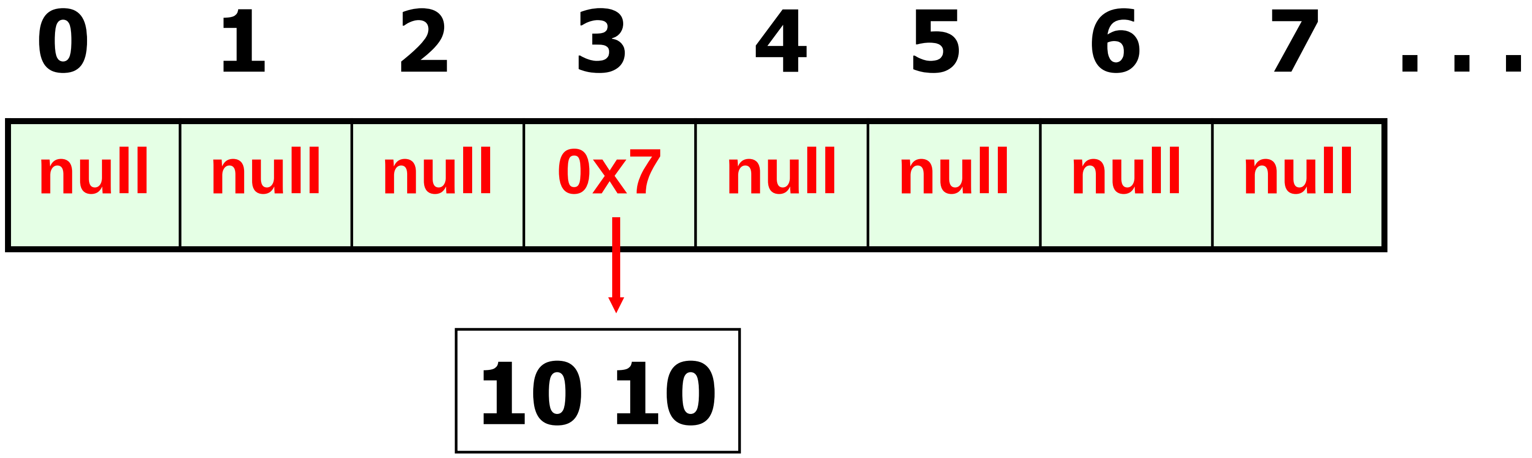
Array of References

```
Monster[] list = new Monster[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] = new Monster( 10, 10 );
```



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

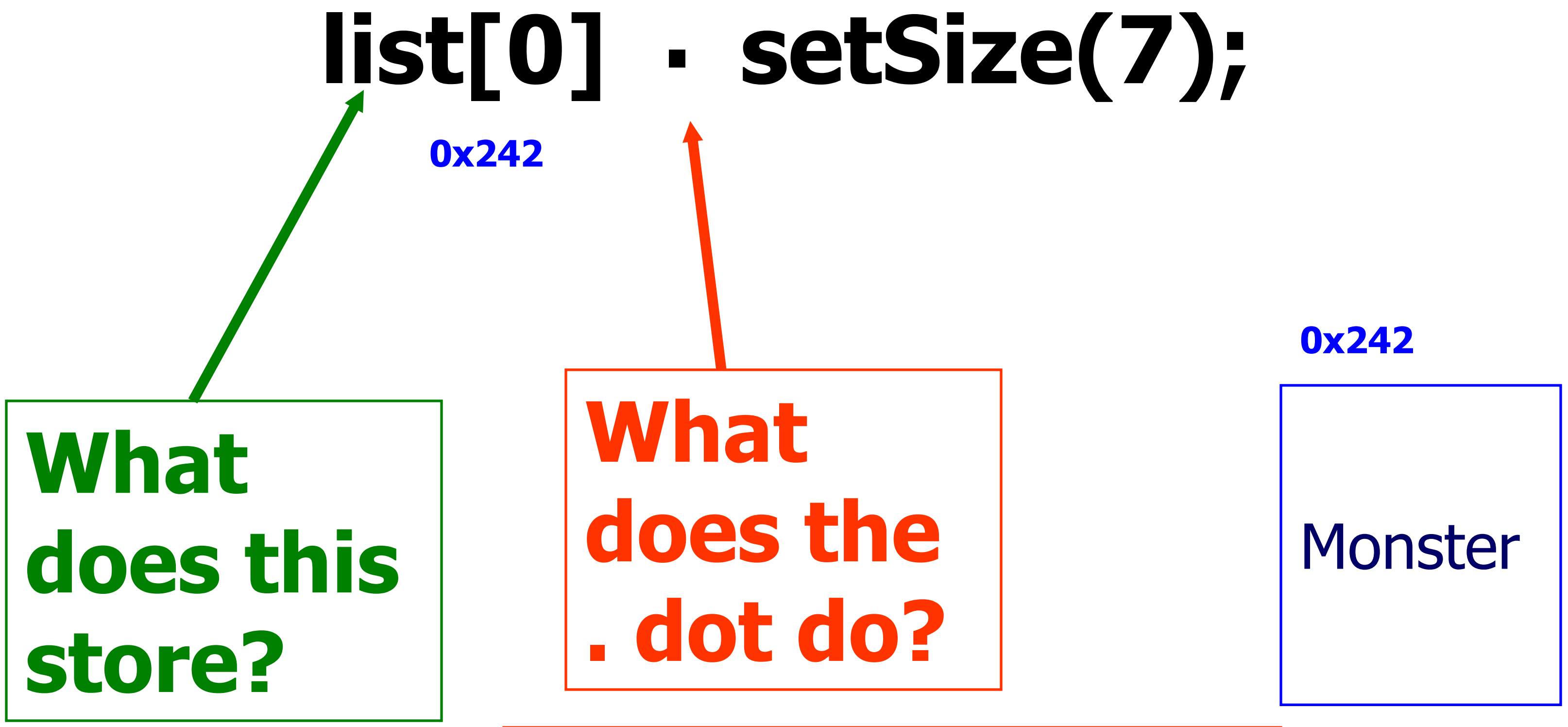
```
Monster[]  list = new Monster[3];
list[0]=new Monster(4);
list[1]=new Monster(9);
list[2]=new Monster(1);

out.println( list[0] );
list[0].setSize(7);

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

OUTPUT

```
4
7
1
```

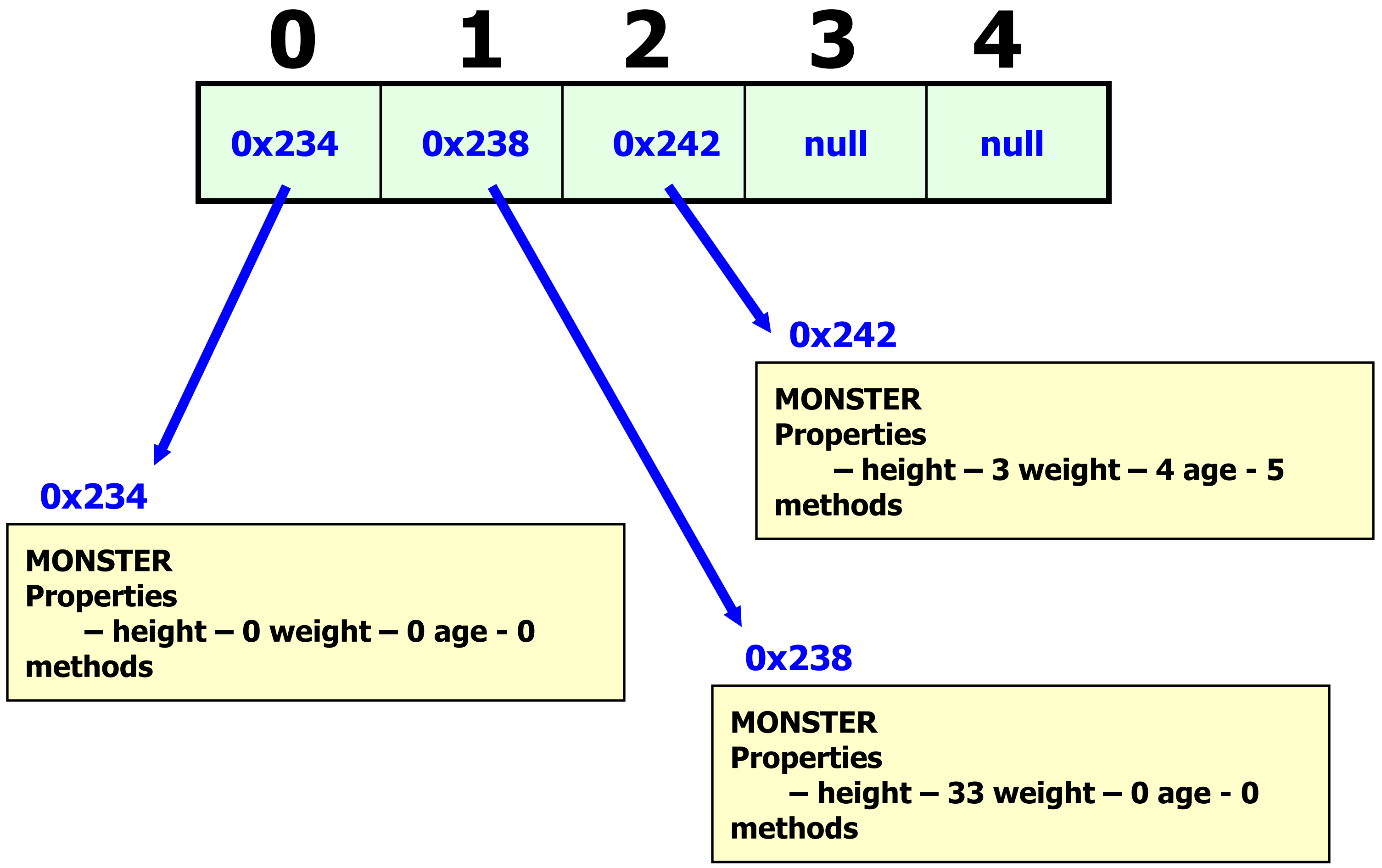
Array of References

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

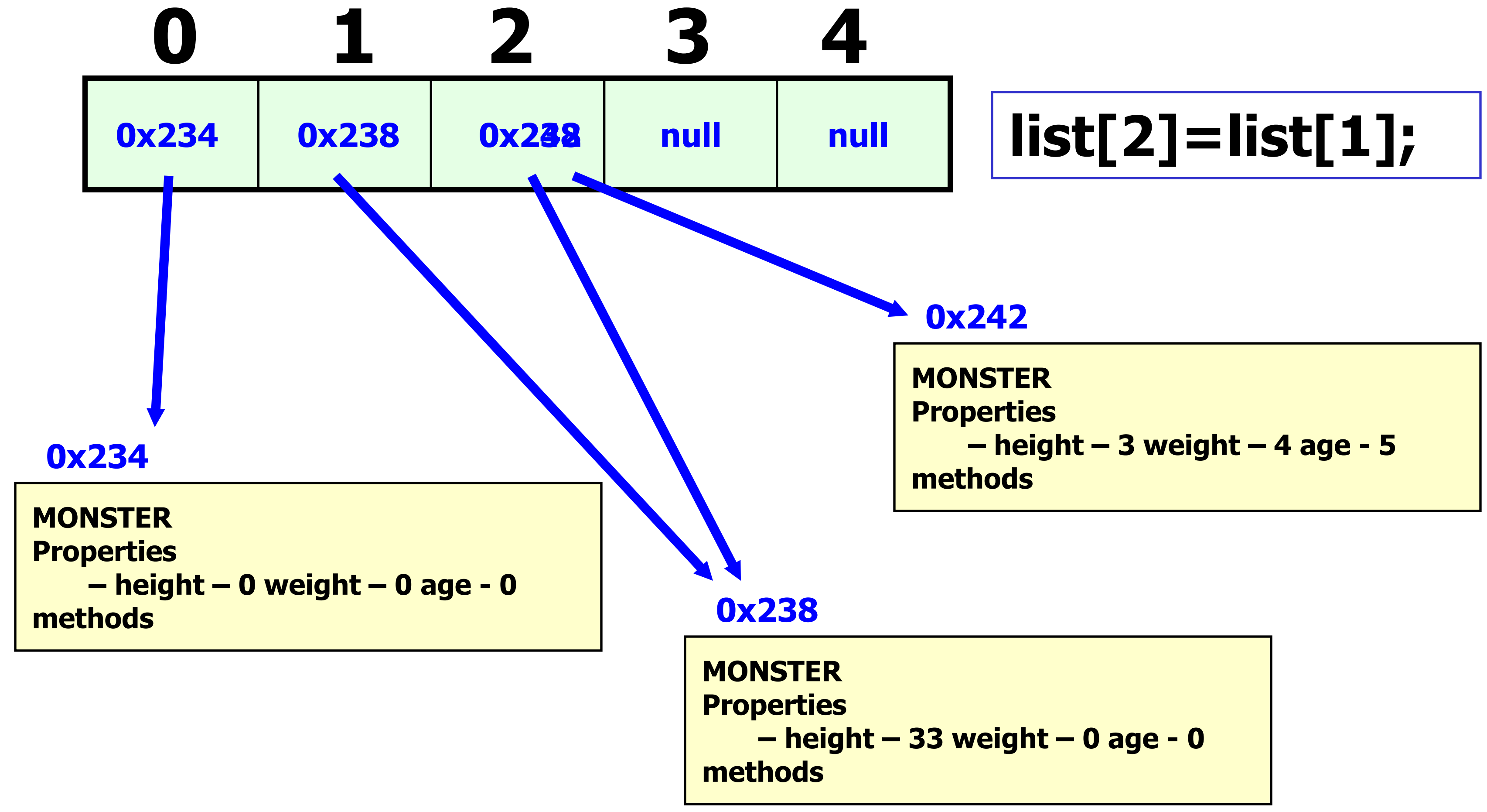


The . dot grants access to the Object at the stored address.

Array of References



Array of References



```
list[2]=list[1];
```

Open arrayofmonsters.java

Instance Variables

Array of References

```
public class Herd
{
   private Creature[]  creatureList;

   public Herd()
   {
     //must size the array

   }

   //other constructors and methods
   //not shown
}
```

Open
creature.java
herd.java
herdrunner.java

String References

References

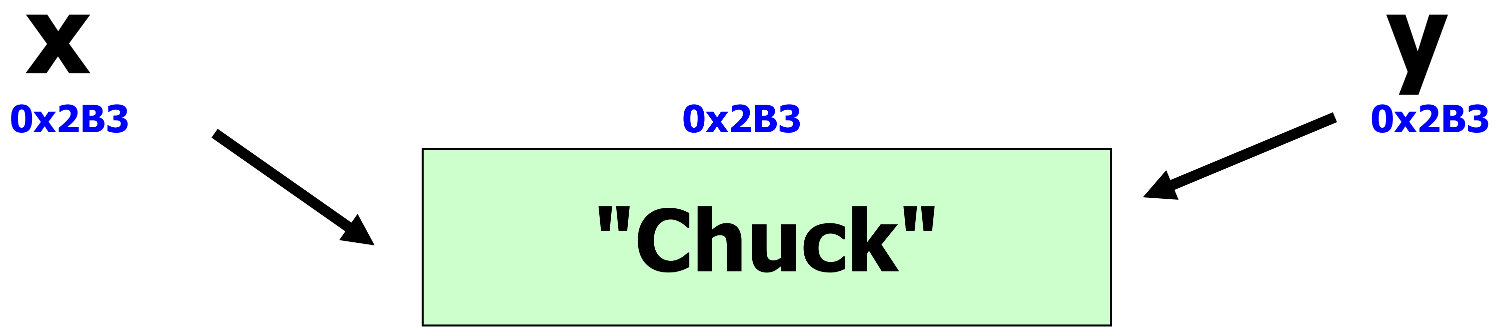
In Java, any variable that refers to an Object is a reference variable.

The variable stores the memory address of the actual Object.

References

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

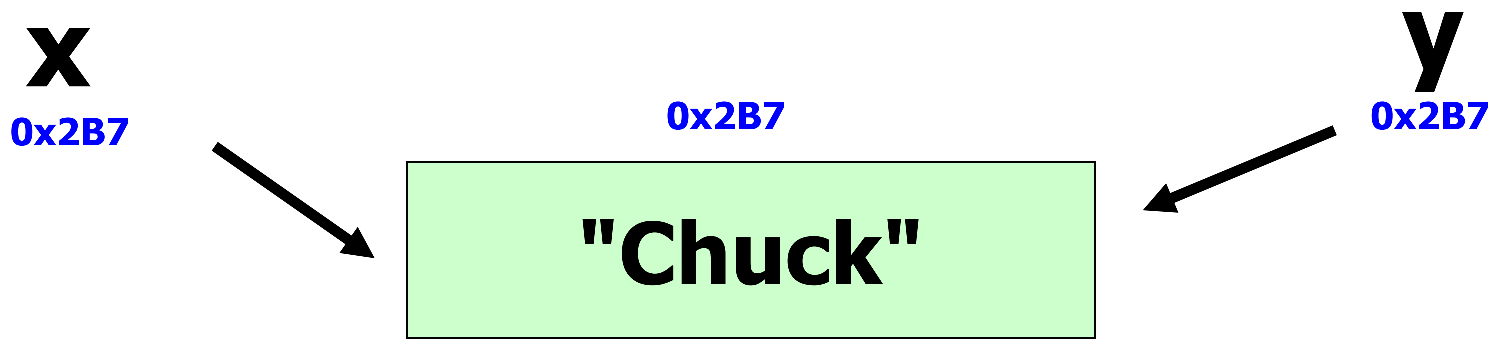
x and y store the same memory address.



References

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

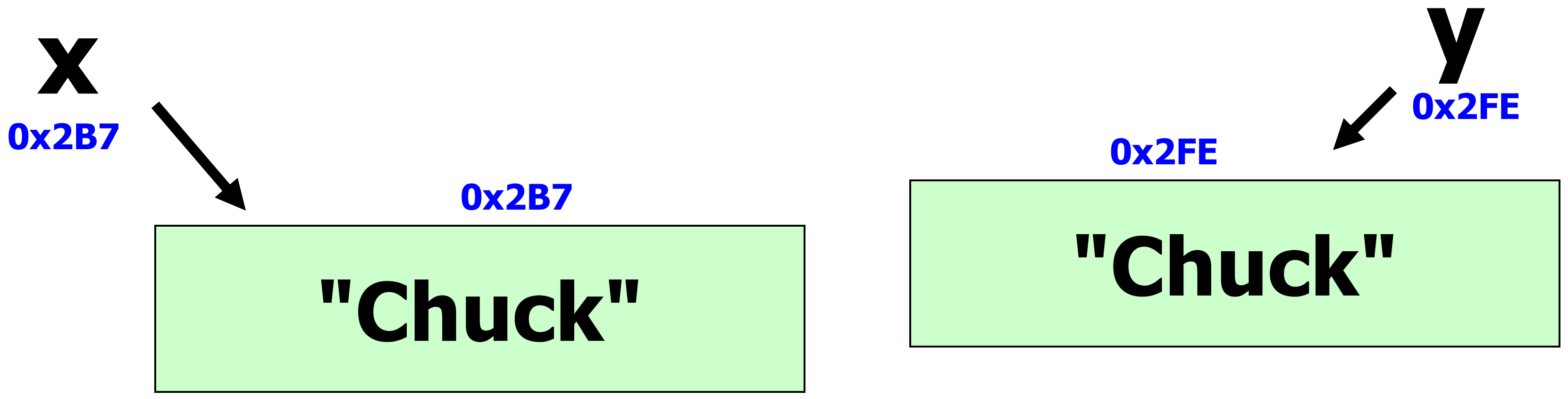
x and y store the same memory address.



References

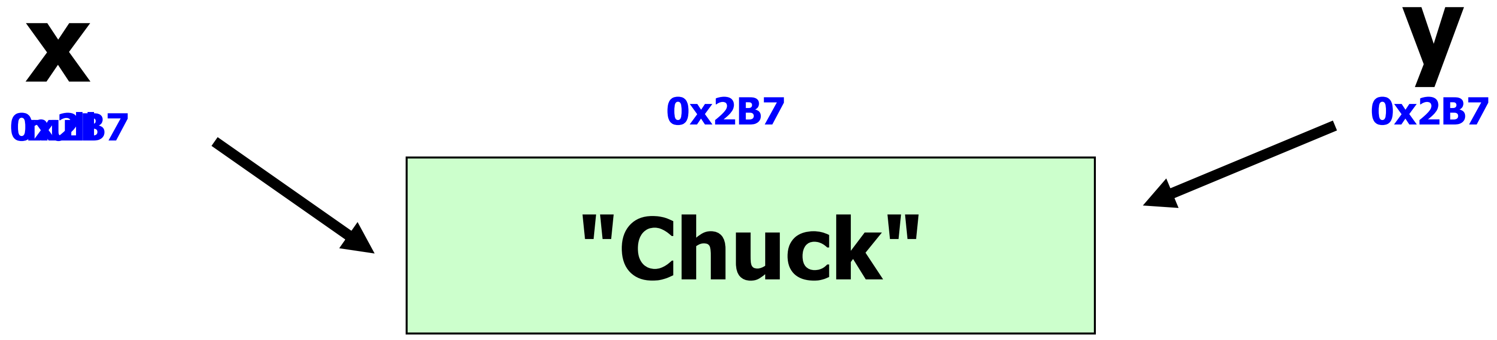
```
String x = new String("Chuck");
String y = new String("Chuck");
```

x and y store different memory addresses.



References

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



open references.java

Array of References

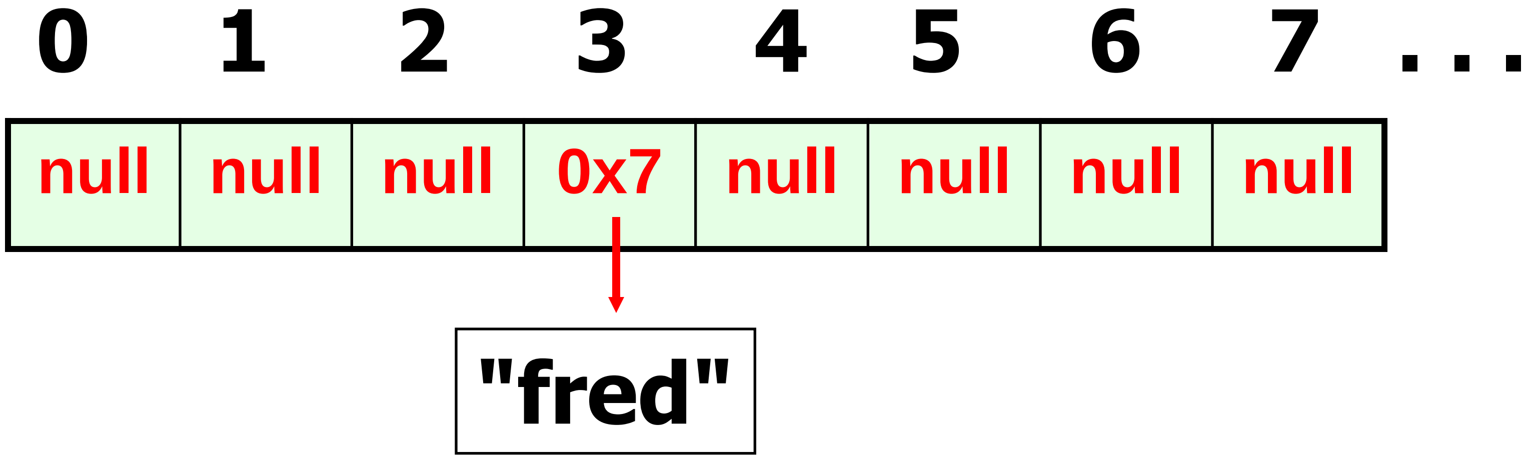
Array of References

```
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 arrayofstrings.java

Start work on the labs