List of References

© A+ Computer Science - www.apluscompsci.com

What is a list?

© A+ Computer Science - www.apluscompsci.com

	Name	Time	Artist	Album	Ge
5	I Dare You to Move	4:08	Switchfoot	Learning to Breathe	
6	I've Been Everywhere	3:20	Johnny Cash	Unchained	
7	Brown Eyed Girl (Single Version)	3:05	Van Morrison	Super Hits	
8	Born to Be Wild	3:31	Steppenwolf	Steppenwolf: All Time Greatest	
9	Magic Carpet Ride	4:28	Steppenwolf	Steppenwolf: All Time Greatest	
10	Crazy (Single Version)	2:42	Patsy Cline	Patsy Cline's Greatest Hits (Ren	
11	Brick House	3:46	The Commodores	20th Century Masters - The Mill	
12	Cleveland Rocks	2:33	The Presidents of the...	Pure Frosting	
13	Chariots of Fire: Main Title Theme	3:32	Carl Davis & Royal Li...	Great Movie Themes	
14	Dueling Banjos (From "Deliverance")	3:11	The Hit Crew	Smash Hit Dramas Movie Theme	
15	Main Theme (From "Superman")	4:12	John Williams	The Music of John Williams - 40	
16	Main Theme (From "Superman")	4:12	John Williams	The Music of John Williams - 40	
17	I've Been Everywhere	3:20	Johnny Cash	Unchained	
18	Born to Be Wild	3:31	Steppenwolf	Steppenwolf: All Time Greatest	

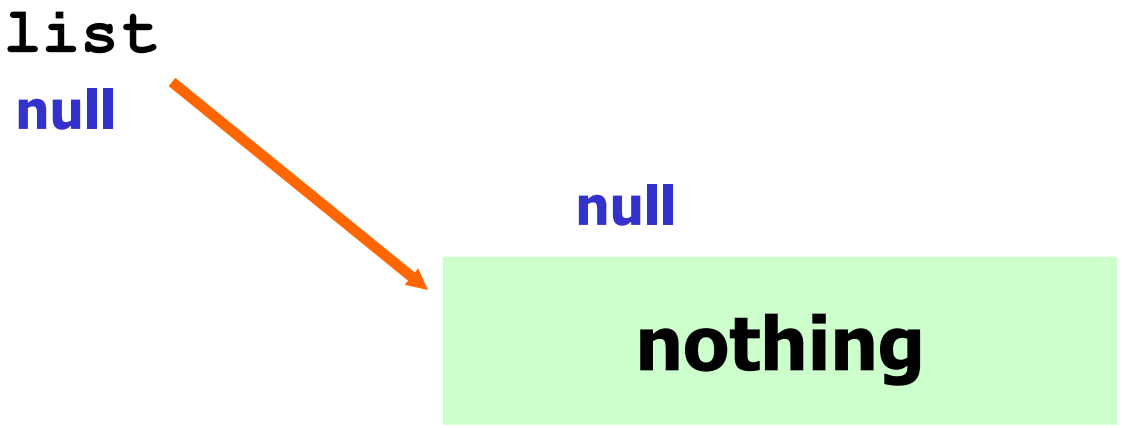
© A+ Computer Science - www.apluscompsci.com

List References

© A+ Computer Science - www.apluscompsci.com

List References

List<SomeClass> list;



list is a reference to some list

© A+ Computer Science - www.apluscompsci.com

ArrayList Instantiation

```
new ArrayList<SomeClass>();
```

0x213

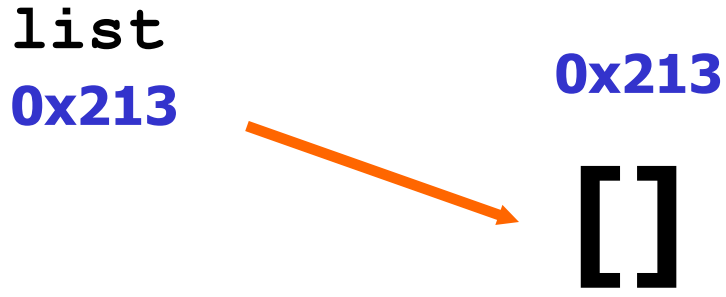
[]

ArrayLists are Objects.

© A+ Computer Science - www.apluscompsci.com

ArrayList

```
List<SomeClass> list;
list = new ArrayList<SomeClass>();
```



list is a reference to an ArrayList.

© A+ Computer Science - www.apluscompsci.com

List Of References

© A+ Computer Science - www.apluscompsci.com

```
public class Dog
{
  private int age;
  private String name;

  public Dog( String n, int a ) {
    age = a;
    name = n;
  }

  public int getAge()   {
    return age;
  }

  public String getName()   {
     return name;
  }

  public String toString()   {
    return "Dog - " + name + " " + age;
  }
}
```

Basic Class

© A+ Computer Science - www.apluscompsci.com

Open Dog.java DogRunner.java

© A+ Computer Science - www.apluscompsci.com

List of References

```
List<Dog> ray;
ray = new ArrayList<Dog>();

ray.add( new Dog( "fred", 11) );
ray.add( new Dog( "ann", 21) );

System.out.println( ray );
```

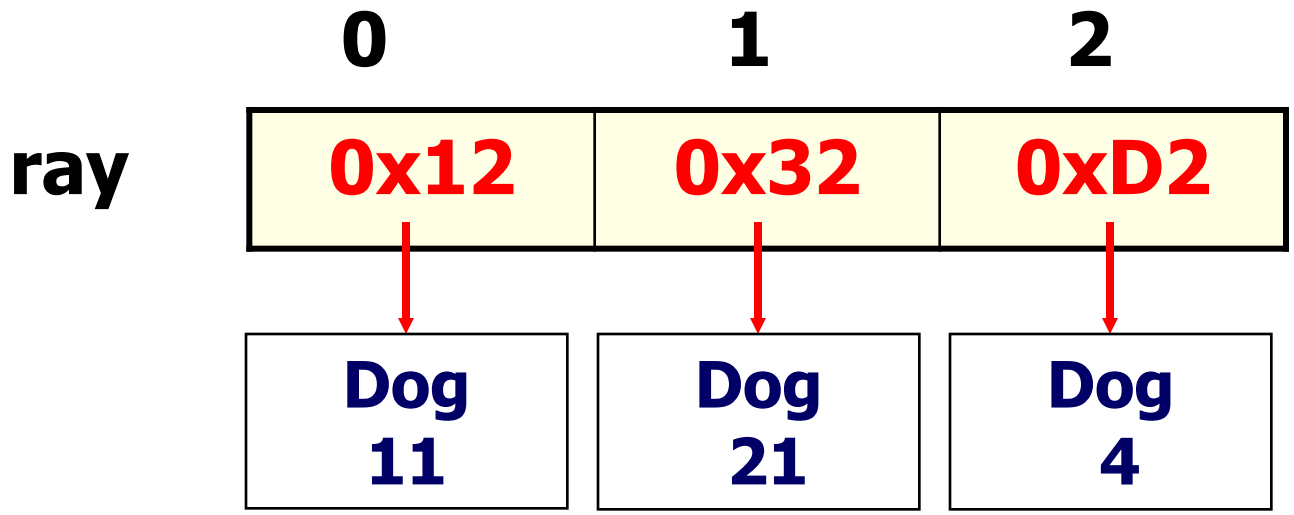
OUTPUT

```
[Dog - fred 11, Dog - ann 21]
```

© A+ Computer Science - www.apluscompsci.com

List of References

```
ray.add( new Dog( "fred", 11) );
ray.add( new Dog( "ann", 21) );
ray.add( new Dog( "bob", 4) );
```



Open DoggiesRunner.java

© A+ Computer Science - www.apluscompsci.com

More Lists of References

© A+ Computer Science - www.apluscompsci.com

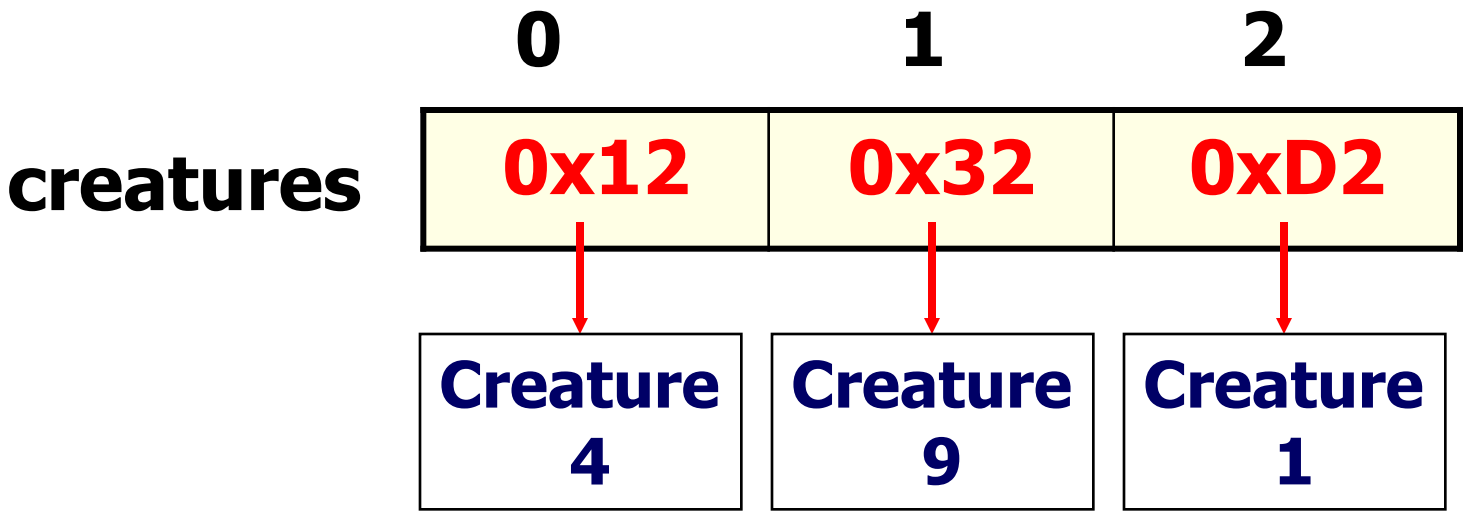
ArrayList w/User Classes

```
public class Creature implements Comparable {
   private int size;

    //checks to see if this Creature is big – size > x
   public boolean isBig()
         //implementation details not show

   public boolean equals(Object obj)
         //implementation details not show

    public int compareTo(Object obj)
         //implementation details not show

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

© A+ Computer Science - www.apluscompsci.com

ArrayList w/User Classes

```
ArrayList<Creature> creatures;
creatures  = new ArrayList<Creature>();
creatures.add(new Creature(4));
creatures.add(new Creature(9));
creatures.add(new Creature(1));
```



© A+ Computer Science - www.apluscompsci.com

ArrayList w/User Classes

```
ArrayList<Creature> creatures;
creatures  = new ArrayList<Creature>();
creatures.add( new Creature(41) );
creatures.add( new Creature(91) );
creatures.add( new Creature(11) );

out.println( creatures.get(0) );

creatures.get(0).setSize(79);
out.println( creatures.get(0) );

out.println( creatures.get(2) );
out.println( creatures.get(1).isBig() );
```

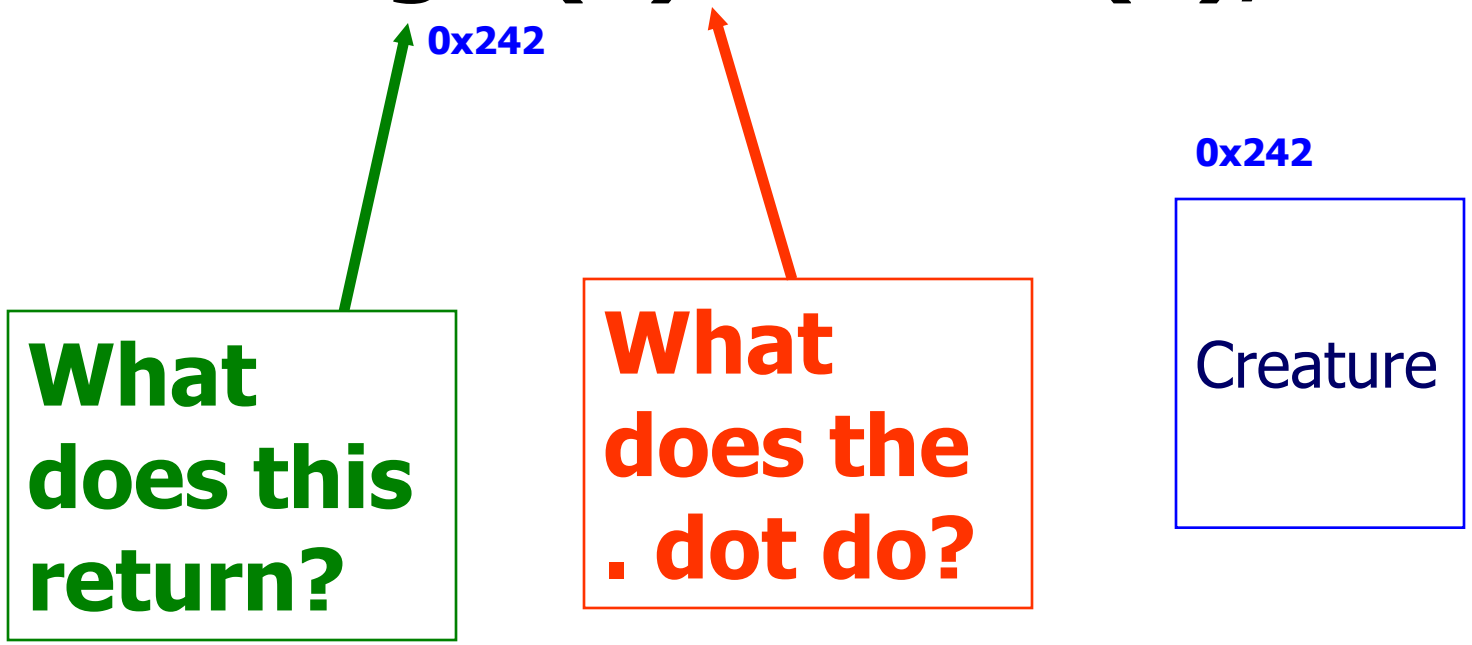
OUTPUT

```
41
79
11
true
```

© A+ Computer Science - www.apluscompsci.com

ArrayList w/User Classes

```
creatures.get(0).setSize(7);
```



© A+ Computer Science - www.apluscompsci.com

ArrayList w/User Classes

```
/* method countBigOnes should return the count of
   big creatures - use the isBig() Creature method
*/
public int countBigOnes()
{
  int cnt = 0;

  //for each loop
     //if statement
          //increase cnt by 1

  return cnt;
}
```

© A+ Computer Science - www.apluscompsci.com

Open creature.java herd.java herdrunner.java

Complete the code

© A+ Computer Science - www.apluscompsci.com

AutoBoxing
AutoUnboxing

© A+ Computer Science - www.apluscompsci.com

Box/Unbox

primitive	object
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean
==	.equals()

© A+ Computer Science - www.apluscompsci.com

Box/Unbox

Before Java 5 added in autoboxing and autounboxing, you had to manually wrap primitives.

```
Integer x = new Integer(98);
int y = 56;
x= new Integer(y);
```

© A+ Computer Science - www.apluscompsci.com

Box/Unbox

Java now wraps automatically.

```
Integer numOne = 99;
Integer numTwo = new Integer(99);
```

```
=99;
=new Integer(99);
```

These two lines are equivalent.

© A+ Computer Science - www.apluscompsci.com

Box/Unbox

Java now wraps automatically.

```
Double numOne = 99.1;
Double numTwo = new Double(99.1);
```

```
=99.1;
=new Double(99.1);
```

These two lines are equivalent.

© A+ Computer Science - www.apluscompsci.com

Box/Unbox

Before Java 5 added in autoboxing and autounboxing, you had to manually unwrap references.

```
Integer ref = new Integer(98);
int y = ref.intValue();
```

© A+ Computer Science - www.apluscompsci.com

Box/Unbox

Java now unwraps automatically.

```
Integer num = new Integer(3);
int prim = num.intValue();
out.println(prim);
prim = num;
out.println(prim);
```

OUTPUT

```
3
3
```

```
prim=num.intValue();
prim=num;
```

These two lines are equivalent.

© A+ Computer Science - www.apluscompsci.com

Box/Unbox

```
Double dub = 9.3;
double prim = dub;
out.println(prim);

int num = 12;
Integer big = num;
out.println(big.compareTo(12));
out.println(big.compareTo(17));
out.println(big.compareTo(10));
```

OUTPUT

```
9.3
0
-1
1
```

© A+ Computer Science - www.apluscompsci.com

Open autoboxunbox.java

© A+ Computer Science - www.apluscompsci.com

new for loop

```
ArrayList<Integer> ray;
ray = new ArrayList<Integer>();

//add some values to ray

int total = 0;
for(Integer num : ray)
{
   //this line shows the AP preferred way
   //it shows the manual retrieval of the primitive
   total = total + num.intValue();

   //the line below accomplishes the same as the line above
   //but, it uses autounboxing to get the primtive value
   //total = total + num;
}
out.println(total);
```

OUTPUT

153

© A+ Computer Science - www.apluscompsci.com

Open foreachloopone.java foreachlooptwo.java

© A+ Computer Science - www.apluscompsci.com

Continue work on Lab 16

© A+ Computer Science - www.apluscompsci.com