

Add stuff here

# What is Magpie?

**Magpie is a lab that focuses on classes, randomness, and Strings.**

**This lab will make sure that you know how to use the String methods substring and indexOf.**

**Both substring and indexOf have multiple forms as these methods have been overloaded.**

Add stuff here

# What is a String?

```
String s = "compsci";
```

	0	1	2	3	4	5	6
s	c	o	m	p	s	c	i

A string is a group of characters.  
The first character in the group is at spot 0.

A String is a group of characters. Strings are used to store words, which can consist of letters, numbers, and symbols.

## **String**

### **Methods from AP CS Subset**

<b>Name</b>	<b>Use</b>
<code>int length()</code>	Returns length of String
<code>int indexOf(String str)</code>	Returns first position of str in the string if found, -1 if not found
<code>String substring(int from)</code>	Returns a substring of the string starting at from to length() – 1
<code>String substring(int from, int to)</code>	Returns a substring of the string starting at from to to – 1

Add stuff here

# substring()

```
String s = "compsci";  
String sub = "";
```

```
sub = s.substring(2);  
out.println(sub);
```

```
sub = s.substring(2,5);  
out.println(sub);
```

```
sub = s.substring(4,6);  
out.println(sub);
```

## OUTPUT

```
mpsci  
mps  
sc
```



	0	1	2	3	4	5	6
s	c	o	m	p	s	c	i

The `String substring()` method returns a `String` containing a section from the original `String`.

# indexOf()



```
String s = "compsci";  
int index = s.indexOf("mp");  
out.println(index);  
index = s.indexOf("c");  
out.println(index);  
index = s.indexOf("x");  
out.println(index);
```

## OUTPUT

```
2  
0  
-1
```

	0	1	2	3	4	5	6
s	c	o	m	p	s	c	i

A+ Computer Science

© A+ Computer Science - www.apluscompsci.com

The `String indexOf()` method looks for a value and returns the spot at which that value is stored. If the value provided is not present in the `String`, `-1` is returned. `-1` would not be a valid spot in the `String` which is why `-1` was chosen as the return value when a value is not found.



Add stuff here



# What is Elevens?

**Elevens is a lab about classes and Lists.**

**List< SomeClass> is a major concept being tested by the Elevens lab.**

**Elevens is a multi-class project that uses a Card and Deck class to simulate the playing of cards.**

Add stuff here

# Lists

```
List<String> ray;  
ray = new ArrayList<String>();  
ray.add("hello");  
ray.add("whoot");  
ray.add("contests");  
out.println(ray.get(0).substring(0, 1));  
out.println(ray.get(2).substring(0, 1));
```

OUTPUT

h  
c

ray stores String references.

In the example above, ray is an ArrayList that stores String references.

```
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 Dog Class

Classes are used to store related methods and variables.

# 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 );
```

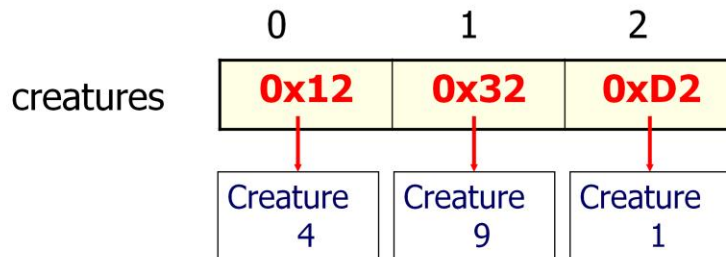
OUTPUT

[Dog - fred 11, Dog - ann 21]

Lists / ArrayLists can store references to objects. This enables each spot in the List / ArrayList to house more than just a single value. Each spot can house multiple variables and methods all of which would be contained in a class.

# List of References

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

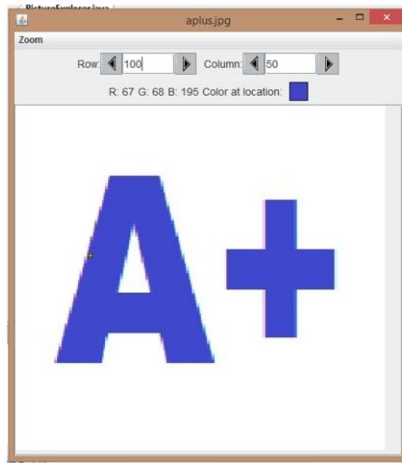


## **ArrayList** frequently used methods

Name	Use
<b>add(item)</b>	adds item to the end of the list
<b>add(spot,item)</b>	adds item at spot – shifts items up->
<b>set(spot,item)</b>	put item at spot <code>z[spot]=item</code>
<b>get(spot)</b>	returns the item at spot <code>return z[spot]</code>
<b>size()</b>	returns the # of items in the list
<b>remove()</b>	removes an item from the list
<b>clear()</b>	removes all items from the list

```
import java.util.ArrayList;
```

# Picture Lab



# What is Picture Lab?

**PictureLab is a lab that focuses on matrices.**

**Matrices are arrays of arrays. The PictureLab will focus heavily on this concept.**

**Matrices can store references. PictureLab will use a matrix of references.**

**Searching matrices is also tested.**



© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

Add stuff here



# What is a matrix?

A matrix is an array of arrays.

```
int[][] mat = new int[3][3];  
mat[0][1]=2;
```

Which  
array?

Which  
spot?

		0	1	2
0	→	0	2	0
1	→	0	0	0
2	→	0	0	0

A+ Computer Science

© A+ Computer Science - www.apluscompsci.com

Each spot in an matrix stores the location/address of an array.

`mat[0]` stores the location / address of a one-dimensional array.

```
mat[0][1]=2;
```

This line sets `mat[0]` spot 1 to 2.

```

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 Dog Class

Classes are used to store related methods and variables.

# Matrix of References

```
Dog[][] herd;  
herd = new Dog[3][3];
```

```
herd[0][0] = new Dog( "fred", 11) ;  
herd[1][2] = new Dog( "ann", 21) ;
```

```
System.out.println( herd[2][2] );  
System.out.println( herd[0][0] );
```

OUTPUT

null  
fred 11

Matrices can store references to objects. This enables each spot in the matrices to house more than just a single value. Each spot can house multiple variables and methods all of which would be contained in a class.

# Searching a Matrix

```
int[][] mat = {{5,7},{5,3,4,6},{0,8,9}};  
int count = 0;  
for( int[] row : mat )  
{  
    for( int num : row )  
    {  
        if( num == 5 )  
            count++;  
    }  
}  
System.out.println("5 count = " + count);
```

**OUTPUT**  
**5 count = 2**

Searching for values in an array or matrix is a common process often tested on the AP exam.

# What is a Picture?



# Start work on the Labs



© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)