

# Strings Return Methods

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

# What is a String?

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

	0	1	2	3	4	5	6
<b>S</b>	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+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

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

# String Constructors

```
String s = "compsci";  
String champ = new String("uilstate");
```

  
**reference  
variable**

  
**object  
instantiation**

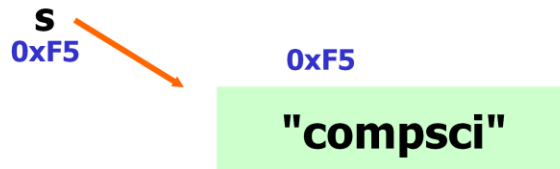
© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

s is a String reference. s is storing the location / memory address of the String Object "compsci";

champ is a String reference. champ is storing the location / memory address of the String Object "uilstate";

# What is a String?

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



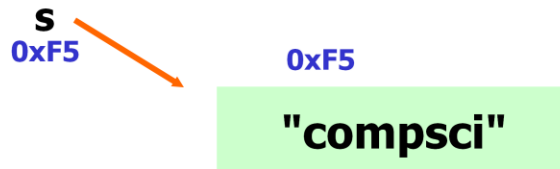
**A reference variable stores the memory address of an object.**

© A+ Computer Science - www.apluscompsci.com

`s` is a String reference. `s` is storing the location / memory address of the String Object `"compsci"`;

# What is a String?

```
String s = new String("compsci");
```



**A reference variable stores the memory address of an object.**

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

`s` is a String reference. `s` is storing the location / memory address of the String Object `"compsci"`;

# **Open basics.java**

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

# Methods

Methods provide / grant access to an object's data / properties.

**String**

instance  
variables /  
data /  
properties

**length( )**

**substring( )**

**indexOf( )**

**toString( )**

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

## **String** frequently used methods

Name	Use
<code>substring(x,y)</code>	returns a section of the string from x to y not including y
<code>substring(x)</code>	returns a section of the string from x to length-1
<code>length()</code>	returns the # of chars
<code>charAt(x)</code>	returns the char at spot x
<code>indexOf(c)</code>	returns the loc of char c in the string, searching from spot 0 to spot length-1
<code>lastIndexOf(c)</code>	returns the loc of char c in the string, searching from spot length-1 to spot 0

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

String is an immutable Object. String cannot be changed. All of the String methods are accessor method. All of the String methods are return methods.



# length()

```
String s = "compsci";  
int len = s.length();  
System.out.println( len );
```

**OUTPUT**  
**7**



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

© A+ Computer Science - www.apluscompsci.com

The String `length()` method returns the character count.  
`length()` looks at the String Object and returns back the number of characters contained.

`compsci` contains 7 characters so a call to `length()` would return 7.

# Return Methods

Return methods perform some action and return a result back.  
`.length()` is a return method.

```
String s = "compsci";  
int len = s.length();  
System.out.println( len );
```

`length()` returns an integer back to the calling location.  
The value returned is then assigned to variable `len`.

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

Return methods typically perform some action then send back a value. Return methods are also used as get methods to retrieve a value from an Object.

# charAt()

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

```
out.print(s.charAt(0) + " ");  
out.print(s.charAt(2) + " ");  
out.println(s.charAt(6));
```

**OUTPUT**

c m i

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

© A+ Computer Science - www.apluscompsci.com

The String `charAt()` method returns the character at the specific spot.

`charAt(0)` would return the character at spot 0.

`charAt(2)` would return the character at spot 2.

# **Open length.java**

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

# **Open charat.java**

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

# substring()

```
String s = "comp sci";  
String sub = "";
```

```
sub = s.substring(3);  
out.println(sub);
```

```
sub = s.substring(0,3);  
out.println(sub);
```

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

## OUTPUT

```
psci  
com  
sci
```



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

© A+ Computer Science - www.apluscompsci.com

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

# 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

© A+ Computer Science - www.apluscompsci.com

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

# **Open substring.java**

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



# 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 - 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.

# indexOf



```
String s = "compSci";  
int index = s.indexOf("pm");  
out.println(index);  
index = s.lastIndexOf("c");  
out.println(index);  
index = s.lastIndexOf("omp");  
out.println(index);
```

## OUTPUT

```
-1  
5  
1
```

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

© 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.

**Open**  
**indexof.java**  
**Complete the code**

# concatenate

```
String one = "computer";  
String two = "-sci";  
String s = one.substring(0,4) + two;  
out.println(s);  
out.println(s.length());
```

**OUTPUT**  
comp-sci  
8

**Concatenate is the process of combining strings together to make a new string.**

© A+ Computer Science - www.apluscompsci.com

It is very common to add strings together make a new string.  
Methods could be used as well as using the plus + operator.

# **Open concatenate.java**

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

# Start work on the labs

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

# **return methods expanded**

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

# Return Methods

Return methods perform some action and return a result back to the **calling location**.

```
int num = keyboard.nextInt();
```

**nextInt()** returns an int back to the calling location.

The value returned is assigned to num.

© A+ Computer Science - www.apluscompsci.com

Return methods typically perform some action then send back a value. Return methods are also used as get methods to retrieve a value from an Object.



# Return Methods

```
Scanner keyboard =  
    new Scanner(System.in);
```

```
int num = keyboard.nextInt();  
out.println(num);
```

num  
1

return  
method

INPUT  
1

OUTPUT  
1

© A+ Computer Science - www.apluscompsci.com

Return methods typically perform some action then send back a value. Return methods are also used as get methods to retrieve a value from an Object.

# Return Methods

```
Scanner keyboard =  
    new Scanner(System.in);
```

```
double num = keyboard.nextDouble();  
out.println(Math.ceil(num));
```

num  
3.45

return  
methods

INPUT  
3.45

OUTPUT  
4.0

© A+ Computer Science - www.apluscompsci.com

Return methods typically perform some action then send back a value. Return methods are also used as get methods to retrieve a value from an Object.

# Return Methods

```
public class ReturnOne
{
    public int twice( int x )  //this is a return method
    {
        return 2*x;
    }
}
```

## OUTPUT

50

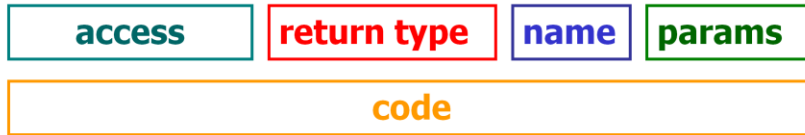
34

```
//code in the main of another class
ReturnOne demo = new ReturnOne();
out.println(demo.twice(25) );
out.println(demo.twice(17) );
```

© A+ Computer Science - www.apluscompsci.com

Method twice is a return method. Method twice takes in parameter x and then sends back x multiplied by 2.

# Return Method



```
public      int      twice( int x )  
{  
    return 2*x;  
}
```

© A+ Computer Science - www.apluscompsci.com

Method twice is a return method. Method twice takes in parameter x and then sends back x multiplied by 2.

# Open returnnone.java

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

# Open returntwo.java

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

# toString

```
class Triangle
{
    private int sideA, sideB, sideC;

    public Triangle(int a, int b, int c)
    {
        sideA=a;
        sideB=b;
        sideC=c;
    }

    public String toString()
    {
        return sideA + " " + sideB + " " + sideC;
    }
}
```

return type

return method



© A+ Computer Science - www.apluscompsci.com

toString() is used to display an Object. print() and println() automatically call toString() when displaying an Object reference. toString() typically sends back all data/properties from an Object as one String.

# **Open toString.java**

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



# **Pieces of the OOP Puzzle Part Three**

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

# constructors

```
public Triangle()  
{  
    sideA=0;  
    sideB=0;  
    sideC=0;  
}
```

**Default  
Constructor**

Constructors are similar to methods.  
Constructors set the properties of an  
object to an initial state.

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

# constructors

```
public Triangle(int a, int b, int c)
{
    sideA=a;
    sideB=b;
    sideC=c;
}
```

**Initialization  
Constructor**

Constructors are similar to methods.  
Constructors set the properties of an  
object to an initial state.

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

# modifier methods

```
public void setSides(int a, int b, int c)
{
    sideA=a;
    sideB=b;
    sideC=c;
}
```

**Modifier methods are methods that change the properties of an object.**

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

# accessor methods

```
public int getSideA()  
{  
    return sideA;  
}
```

**Accessor methods are methods that retrieve or grant access to the properties of an object, but do not make any changes.**

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

# accessor methods

```
public String toString()  
{  
    return "" + getSideA() + " " + sideB + " " + sideC;  
}
```

**Accessor methods are methods that retrieve or grant access to the properties of an object, but do not make any changes.**

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

# **encapsulation**

**All data members should have private access. The public constructors, accessor methods, and modifier methods should be used to manipulate the data. All data is tucked away nicely inside the class.**

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

# encapsulation

The public methods give you access to an object's private data / properties.

**Class/  
Object**

private data /  
instance variables /  
properties

getIt( )

setIt( )

toString( )

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



**Open**  
**triangle.java**  
**trianglerunner.java**

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

# Continue work on the labs

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