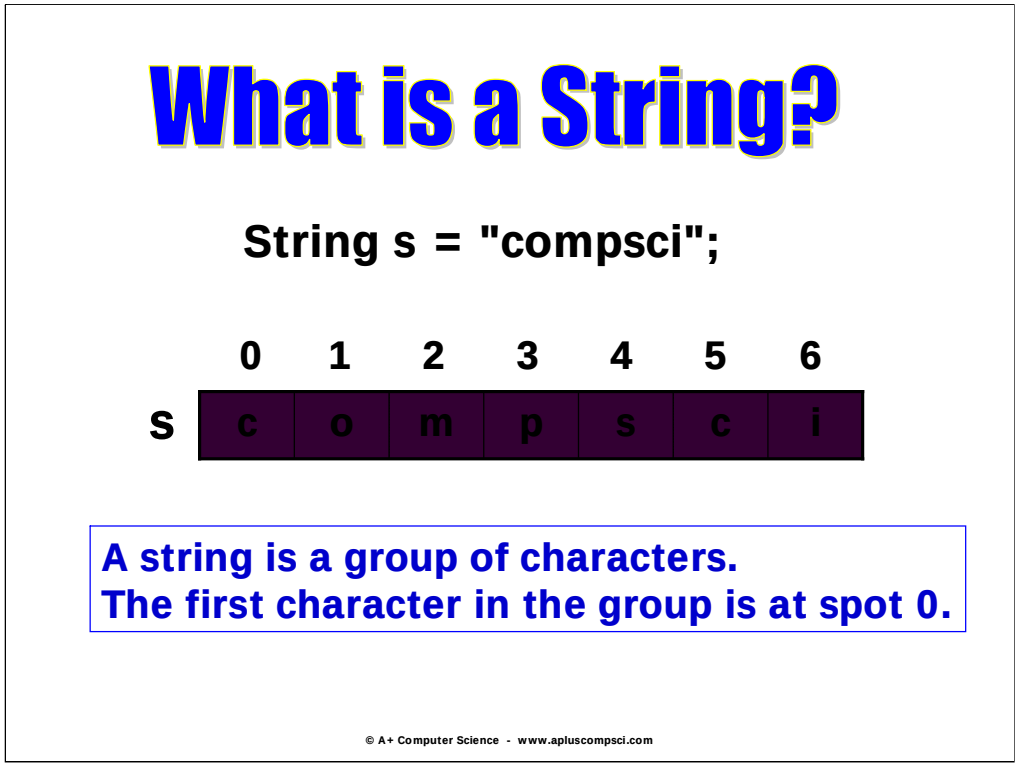
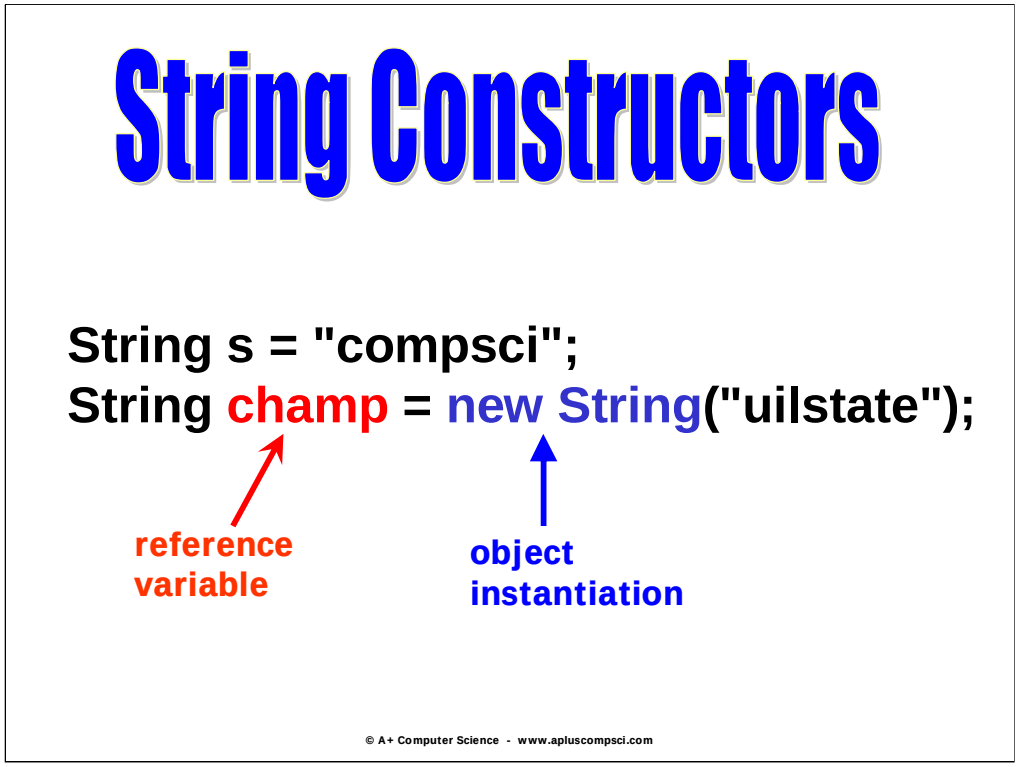
Strings
Return Methods

Lab 04

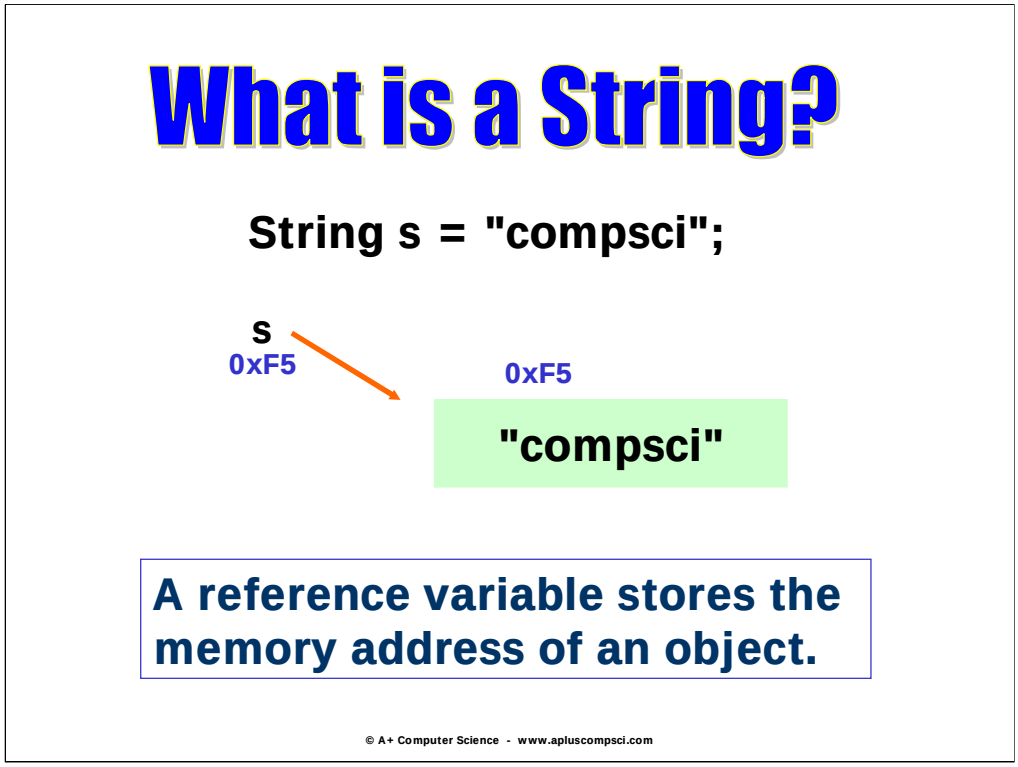


A String is a group of characters. Strings are used to store words, which can consist of letters, numbers, and symbols.



`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"`;

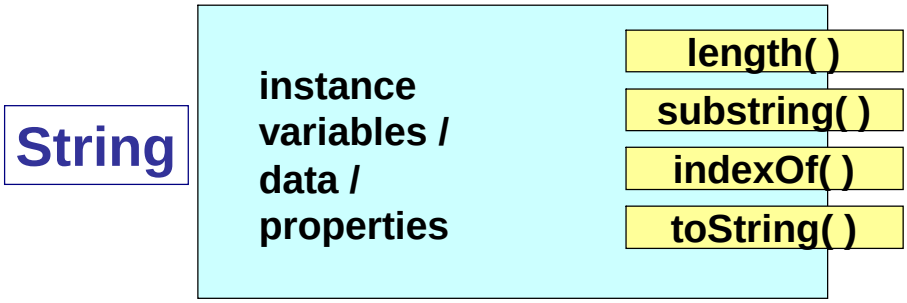


`s` is a String reference. `s` is storing the location / memory address of the String Object `"compsci"`;

Open basics.java

Methods

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



String

frequently used methods

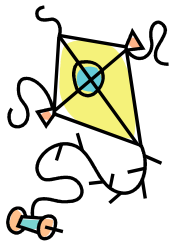
Name	Use
substring(x,y)	returns a section of the string from x to y not including y
substring(x)	returns a section of the string from x to length-1
length()	returns the # of chars
charAt(x)	returns the char at spot x
indexOf(c)	returns the loc of char c in the string, searching from spot 0 to spot length-1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

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()

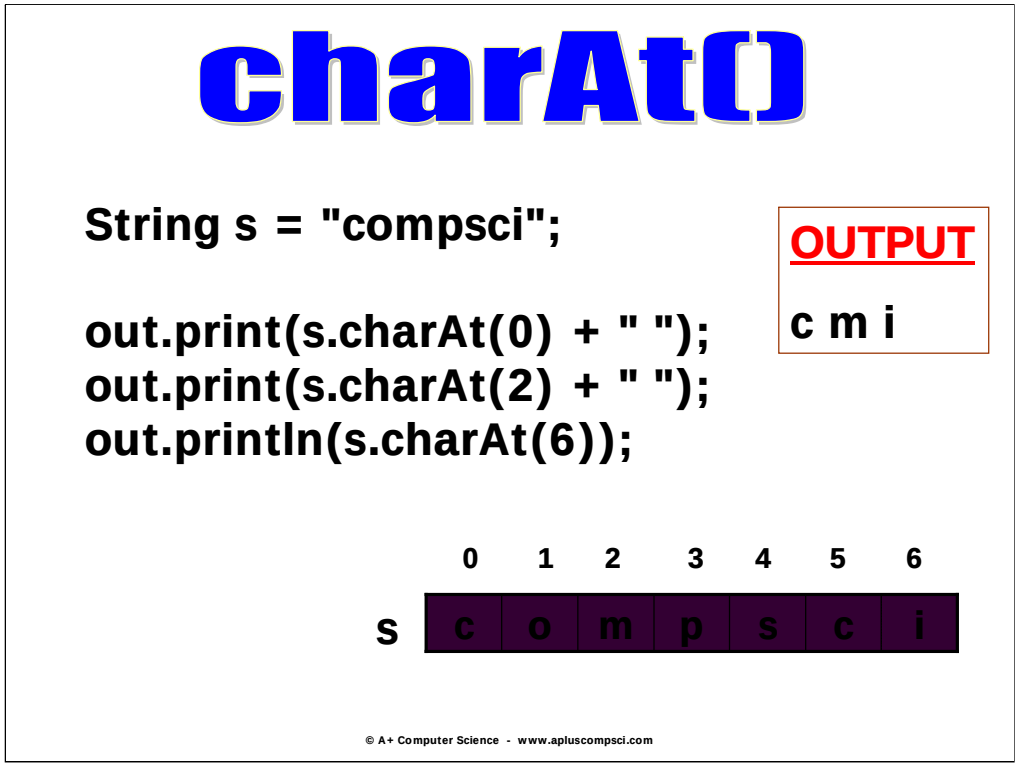
```
String s = "compsci";
out.println(s.length());
```

OUTPUT
7



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

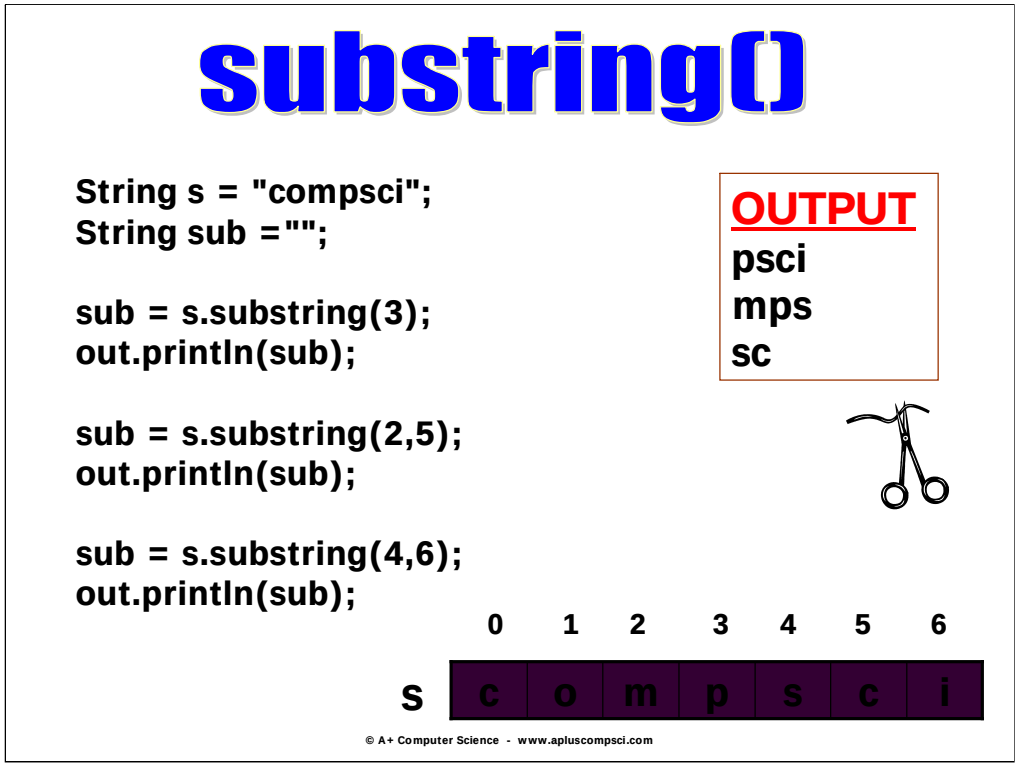
`compsci` contains 7 characters so a call to `length()` would return 7.



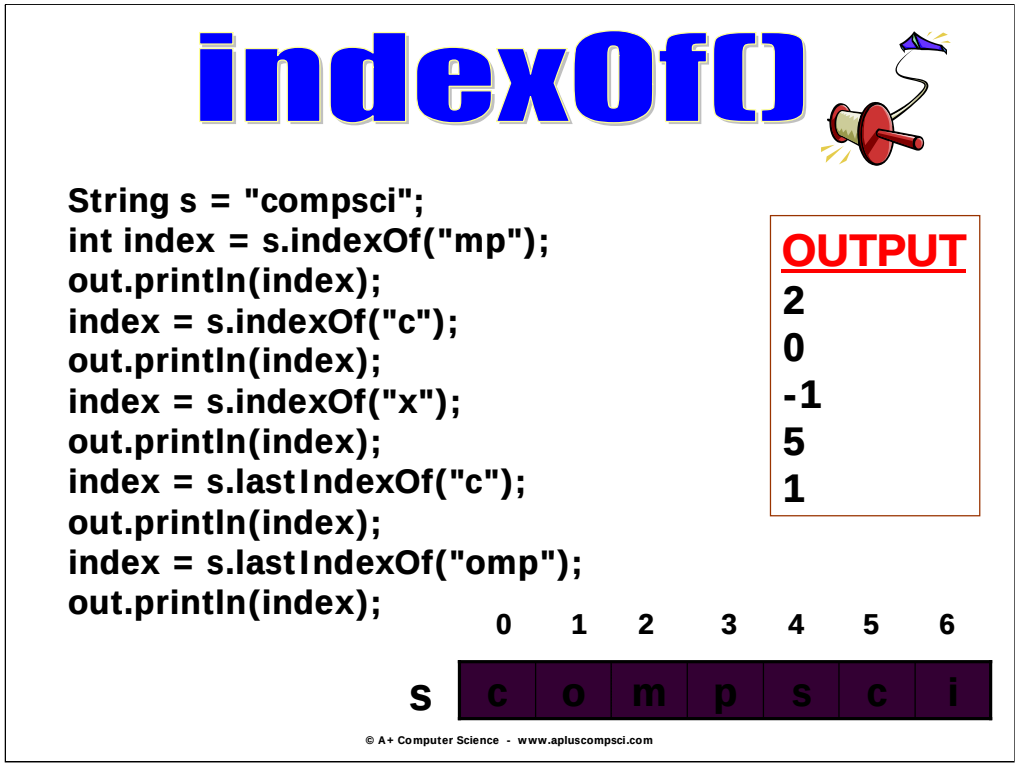
The String `charAt()` method returns the character at the specific spot.

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

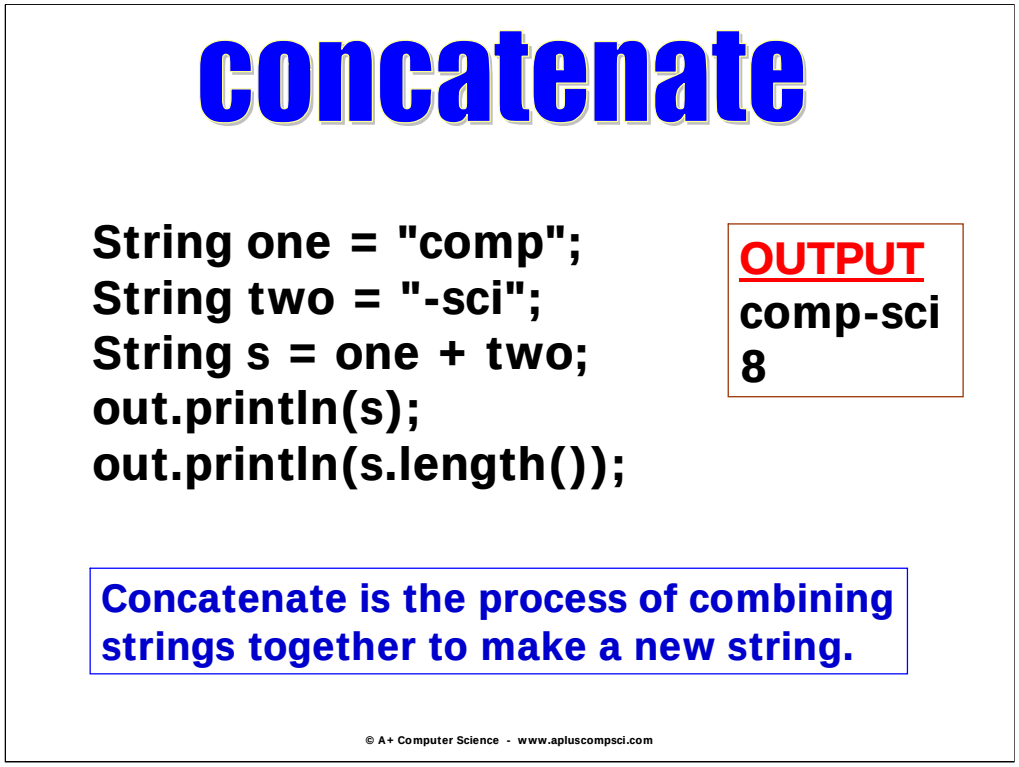
`charAt(2)` would return the character at spot 2.



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



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.



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

Open length.java

Open charat.java

Open substring.java

Open indexof.java

Open concatenate.java

Start work on Lab 04

return methods expanded

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.

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);
```

INPUT
1

OUTPUT
1

num
1

return method

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));
```

INPUT
3.45

OUTPUT
4.0

num
3.45

return methods

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

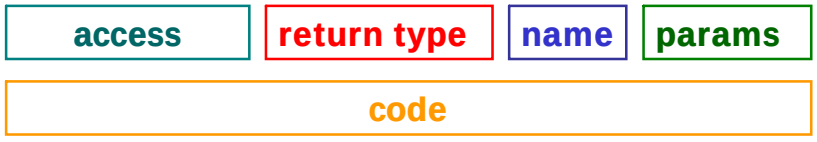
```
public class ReturnOne
{
   public int twice( int x )    //this is a return method
   {
      return 2*x;
   }
}
```

OUTPUT
50
34

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


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;
}
```


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

Open returnone.java

Open returntwo.java

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

`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

Pieces of the OOP Puzzle Part Three

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.

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.

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.

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.

accessor methods

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

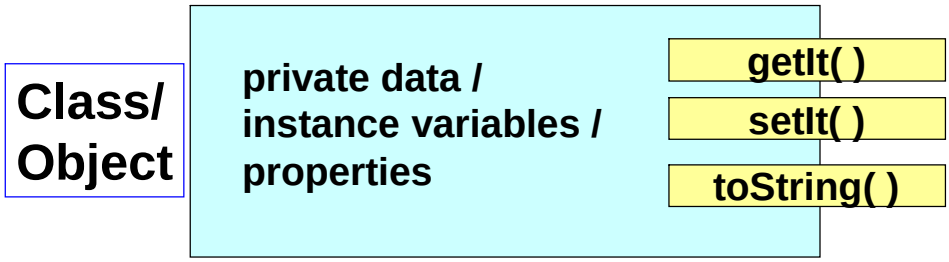
Accessor methods are methods that retrieve or grant access to the properties of an object, but do not make any changes.

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.

encapsulation

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



Open triangle.java triangleRunner.java

Continue work on Lab 04