

try

```
double a = 1/3;  
int b = 1/3;  
double c = 1./3.;  
System.out.println("a = " + a);  
System.out.println("b = " + b);  
System.out.println("b = " + (Math.round(b*100)/100));  
System.out.println("b = " + (Math.round(b*100)/100.0));  
System.out.println("c = " + c);
```

## Objects

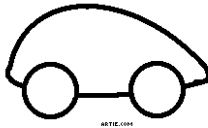
## Getting classy

- Your current job
  - Gain experience creating and manipulating objects from the standard Java types
- Why
  - Prepares you for defining your own classes and creating and manipulating the objects of those classes

## Values versus objects

- Numbers
  - Have values but they do *not* have behaviors
- Objects
  - Have attributes and behaviors

# Objects



# Some other Java object types

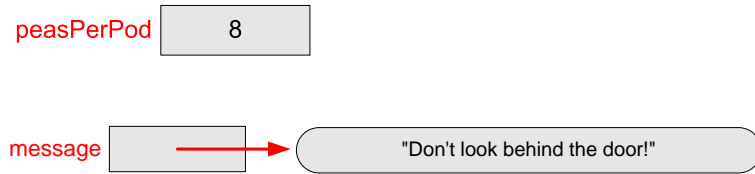
- String
- Rectangle
- Color
- JFrame
  
- **Scanner** (J2SE1.5)
  - Scanner inputNumber = new Scanner(System.in);  
double weight = inputNumber.nextDouble();
  - In J2SE 1.4.2 or below, you need to use:
    - // set up input stream  
BufferedReader stdin = new BufferedReader(  
new InputStreamReader(System.in));
  
    - // get person's characteristics  
System.out.print("Enter weight (lbs): ");  
double weight = Double.parseDouble(stdin.readLine());

The screenshot shows a web browser window with the address bar containing `http://java.sun.com/j2se/1.4.2/docs/api/`. The page title is "Java™ 2 Platform Std. Ed. v1.4.2". The main content area displays the documentation for the `String` class, including navigation links, package information, and the class signature: `public final class String`. The browser's taskbar at the bottom shows the start button, Netscape, Internet Explorer, and Microsoft PowerPoint.

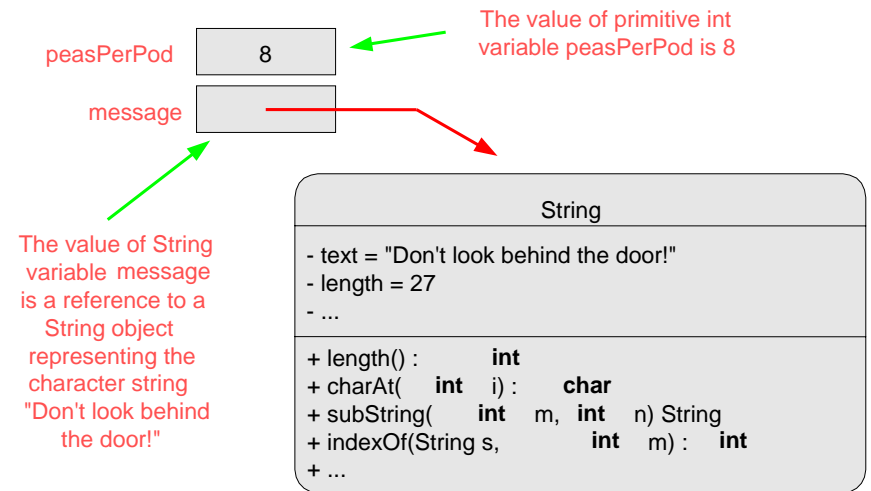
# Consider

- Statements
  - `int peasPerPod = 8;`
  - `String message = "Don't look behind the door!"`
  
- How should we represent these definitions according to the notions of Java?

## Shorthand representation



## Representation

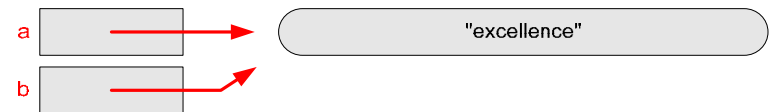


## Examples

- Consider  
`String a = "excellence";`  
`String b = a;`
- What is the representation?

## Examples

- Consider  
`String a = "excellence";`  
`String b = a;`
- What is the representation?



## Uninitialized versus null

- Consider  
`String dayOfWeek;`

- What is the representation?

## Uninitialized versus null

- Consider  
`String dayOfWeek;`

- What is the representation?

dayOfWeek 

## Uninitialized versus null

- Consider  
`String fontName = null;`

- What is the representation?

## Uninitialized versus null

- Consider  
`String fontName = null;`

- What is the representation?

fontName 

## Difference between NULL and UNINITIALIZED references

```
String s1 = null;
```

```
String s2;
```

```
String s3 = s1;
```

*What is s3?*

```
s3 = null    !!
```

```
String s4 = s2;
```

*What is s4?*

*This is an illegal statement!*

## Assignment

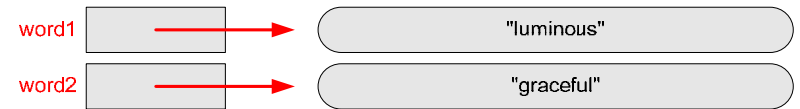
- Consider

```
String word1 = "luminous";
```

```
String word2 = "graceful";
```

```
word1 = word2;
```

- Initial representation



## Assignment

- Consider

```
String word1 = "luminous";
```

```
String word2 = "graceful";
```

```
word1 = word2;
```

- After assignment



What about string "luminous"?

## Using objects

- Consider

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

```
System.out.print("Enter your account name: ");
```

```
String response = stdin.nextLine();
```

## Using objects

- Consider

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

```
System.out.print("Enter your account name: ");
```

```
String response = stdIn.nextLine();
```

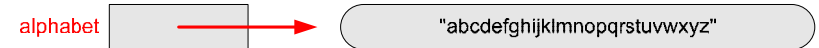


## String representation

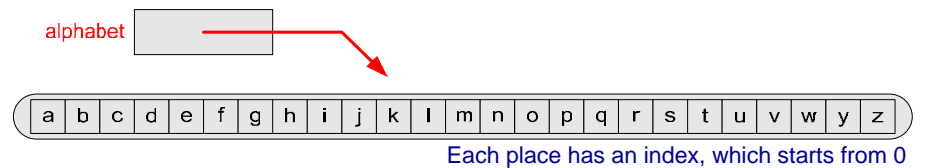
- Consider

```
- String alphabet = "abcdefghijklmnopqrstuvwxyz";
```

- Standard shorthand representation



- Truer representation



## String representation

- Consider

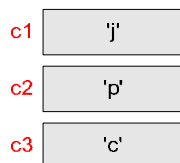
```
- String alphabet = "abcdefghijklmnopqrstuvwxyz";
```

```
- char c1 = alphabet.charAt(9);
```

```
- char c2 = alphabet.charAt(15);
```

```
- char c3 = alphabet.charAt(2);
```

- What are the values of c1, c2, and c3? Why?



## Strings

```
String s = "I love Java";
```

```
s += "programming";
```

```
s += "!";
```

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

*I love Java programming!*

## More String methods

- Consider

```
String weddingDate = "August 21, 1976";
String month = weddingDate.substring(0, 6);
System.out.println("Month is " + month + ".");
```
- What is the output?

## More String methods

- Consider

```
String weddingDate = "August 21, 1976";
String month = weddingDate.substring(0, 6);
System.out.println("Month is " + month + ".");
```
- What is the output?  
Month is August.

The substring begins at the specified beginIndex and extends to the character at index endIndex - 1

## More String methods

- Consider

```
String fruit = "banana";
String searchString = "an";
int n1 = fruit.indexOf(searchString, 0);
int n2 = fruit.indexOf(searchString, n1 + 1);
int n3 = fruit.indexOf(searchString, n2 + 1);

System.out.println("First search: " + n1);
System.out.println("Second search: " + n2);
System.out.println("Third search: " + n3);
```
- What is the output?

## More String methods

- Consider

```
String fruit = "banana";
String searchString = "an";
int n1 = fruit.indexOf(searchString, 0);
int n2 = fruit.indexOf(searchString, n1 + 1);
int n3 = fruit.indexOf(searchString, n2 + 1);

System.out.println("First search: " + n1);
System.out.println("Second search: " + n2);
System.out.println("Third search: " + n3);
```
- What is the output?  
First search: 1  
Second search: 3  
Third search: -1

## More String methods

- Consider

```
int v1 = -12;  
double v2 = 3.14;  
char v3 = 'a';  
String s1 = String.valueOf(v1);  
String s2 = String.valueOf(v2);  
String s3 = String.valueOf(v3);
```

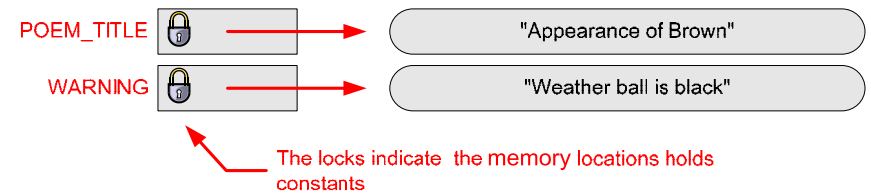


## Final variables

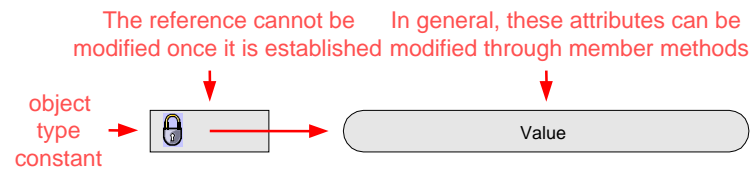
- Consider

```
final String POEM_TITLE = "Appearance of Brown";  
final String WARNING = "Weather ball is black";
```

- What is the representation?



## Final variables



## Operating standard geometries

- Example: Rectangle



The screenshot shows the Java API documentation for the `Rectangle` class in the `java.awt` package. The page title is "Class Rectangle" and it lists the package as `java.awt`. The class is noted to extend `Rectangle2D` and implement `Shape` and `Serializable`. A brief description states: "A Rectangle specifies an area in a coordinate space that is enclosed by the Rectangle object's top-left point (x, y) in the coordinate space, its width, and its height." The URL at the bottom is <http://java.sun.com/j2se/1.4.2/docs/api/index.html>.

## Using Objects: Rectangle

```
int x = 3;
int y = 4;
int width = 5;
int height = 2;
Rectangle r = new Rectangle(x, y, width, height);
```

The first two parameters of the Rectangle constructor specify the position of the upper-left-hand corner of the new Rectangle

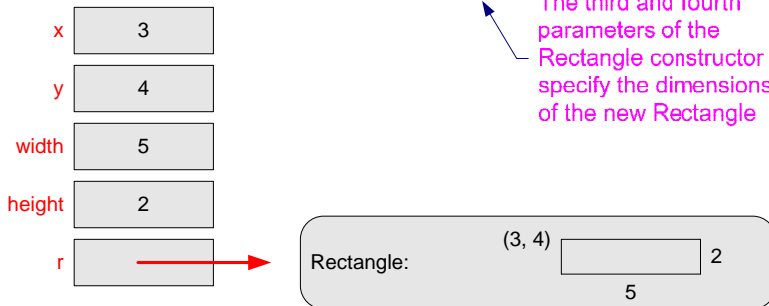
The third and fourth parameters of the Rectangle constructor specify the dimensions of the new Rectangle

## Rectangle

```
int x = 3;
int y = 4;
int width = 5;
int height = 2;
Rectangle r = new Rectangle(x, y, width, height);
```

The first two parameters of the Rectangle constructor specify the position of the upper-left-hand corner of the new Rectangle

The third and fourth parameters of the Rectangle constructor specify the dimensions of the new Rectangle



## Rectangle

- Consider

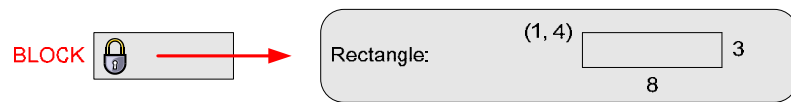
```
final Rectangle BLOCK = new Rectangle(6, 9, 4, 2);
BLOCK.setLocation(1, 4);
BLOCK.resize(8, 3);
```



## Rectangle

- Consider

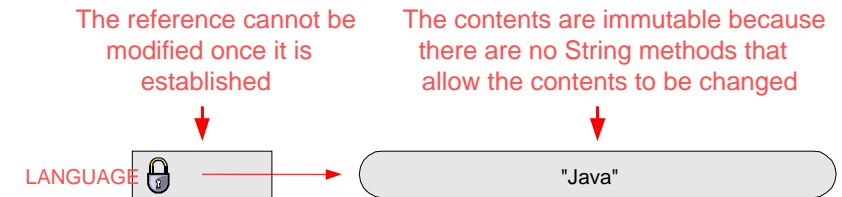
```
final Rectangle BLOCK = new Rectangle(6, 9, 4, 2);  
BLOCK.setLocation(1, 4);  
BLOCK.resize(8, 3);
```



## Final variables

- Consider

```
final String LANGUAGE = "Java";
```



## Assignment (Due Feb 6 before noon)

- Write a program that defines a `Rectangle` object, and then the program can accept user's inputs to change its location and size, after it is resized, output its new location, new size, and then compute and output its area.
- Write a program that prompts for and accepts a telephone number in the form "ddddddddd", and then output the new telephone number in the form "+0852(dddd)dddd". For example, if you input `26098045`, the output should be `+0852(2609)8045`
- Write a program that prompts for and accepts a date in the form of "day/month/year" and output in the form of "month, day, year". For example, if you input `30/January/2007`, the output should be like:  
The input is `"30/January/2007"`, the output is `"January, 30, 2007"`.