



Practice with Objects Strings, Scanners, etc.







February 7, 2012

CMPSCI 121, Spring 2012

Introduction to Problem Solving with Computers

Prof. Learned-Miller

Assignments

Requirement Status	Assignment	Due Date
	 eBook - Chapter 3: Classes, Strings, and I.O.	2/7/2012 11:30 PM
	Chapter 3 Exercises	2/8/2012 11:30 PM
	Bank Accounts	2/9/2012 11:30 PM
	eBook - Chapter 4: Looping and Conditionals	2/14/2012 11:30 PM
	Chapter 4 Exercises	2/16/2013 11:30 PM

Grading of assignments

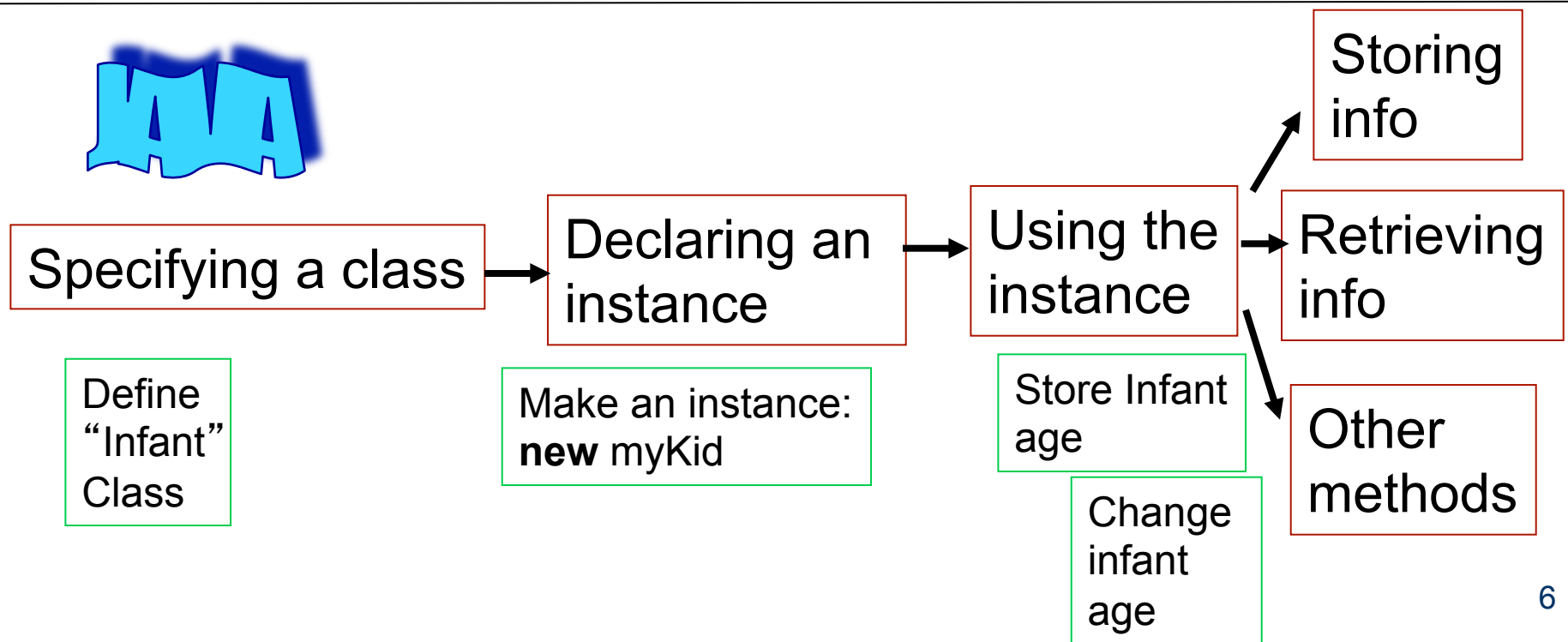
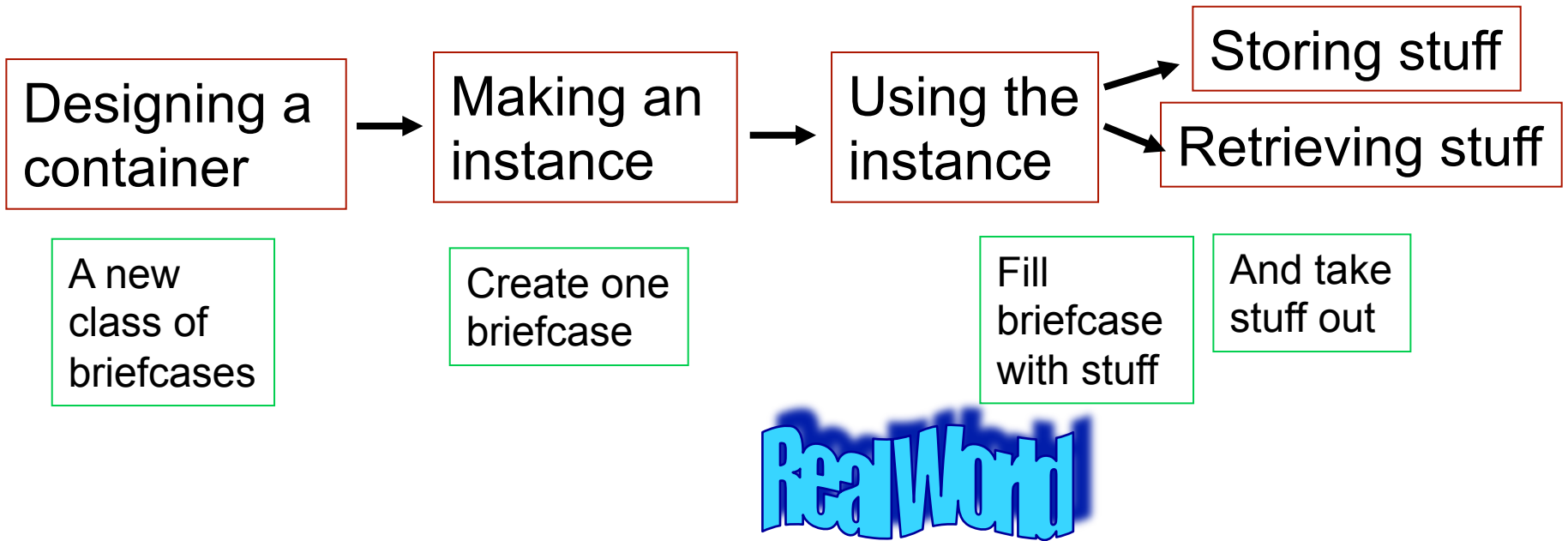
Bean Shell problems

Today

- review of classes, objects
- types
- more on the String class
- import
- APIs
- the Scanner class

Review of objects

- Class: a type of container
(Gucci briefcases for 2012).
- Instance: one example of a class
(my briefcase)
- Class definition: The part of the program that lays out the structure of a class
(blueprint for the class)



Remember the actors involved in the play...

- The design of a briefcase may be done at a different time and place than the construction of the briefcase.
- The person who constructs a briefcase may be a different person than the person who designed the briefcase.
- **Making a design** and **making an instance** are two fundamentally different processes!


```
public class MacSong
{
    public static void main(String[] args)
    {
        MacChorus chorus = new MacChorus();
        MacVerse pig = new MacVerse("pig", "oink");
        MacVerse dog = new MacVerse("dog", "woof");
        chorus.showChorus();
        pig.verse();
        chorus.showChorus();
        chorus.showChorus();
        dog.verse();
        chorus.showChorus();
    }
}
```

```

public class MacVerse
{
    // animal name in verse
    private String name;
    // animal noise in verse
    private String noise;

    public MacVerse(String animalName, String animalNoise)
    {
        name = animalName;
        noise = animalNoise;
    }

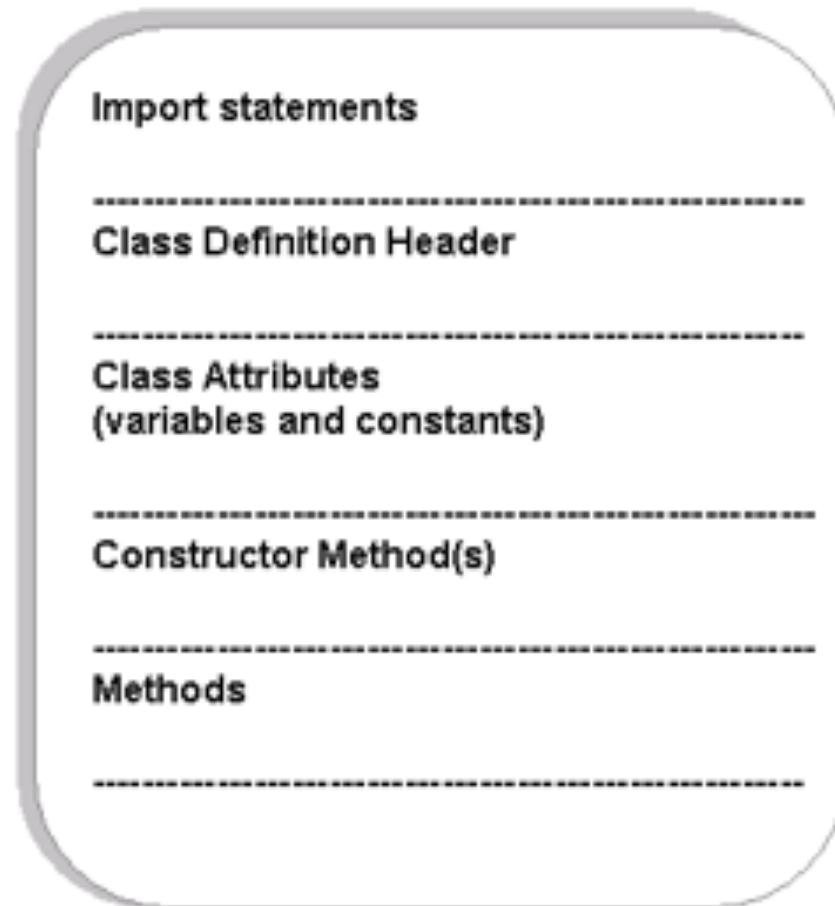
    public String getName()
    {
        return name;
    }

    public String getNoise()
    {
        return noise;
    }

    public void verse()
    {
        System.out.println("And on that farm he had a " + name);
        System.out.println("ei ei o");
        System.out.println("With an " + noise + " " + noise + " here");
        System.out.println("And a " + noise + " " + noise + " there");
        System.out.println("Here a " + noise + " there a " + noise);
        System.out.println("Everywhere a " + noise + " " + noise);
    }
}

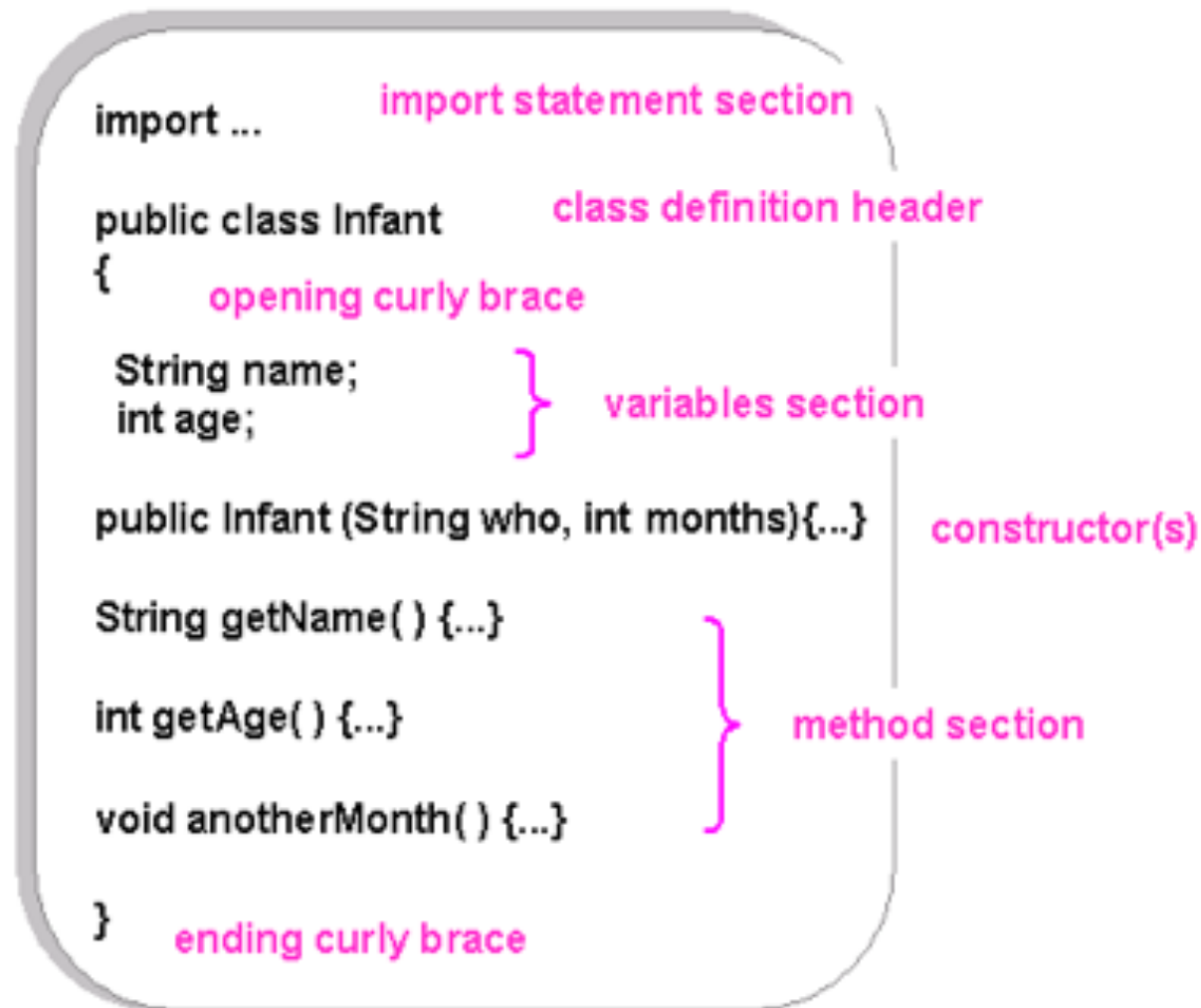
```

Structure of a class definition



Java Class Anatomy

Structure of a class definition



```

public class MacVerse
{
    // animal name in verse
    private String name;
    // animal noise in verse
    private String noise;

    public MacVerse(String animalName, String animalNoise)
    {
        name = animalName;
        noise = animalNoise;
    }

    public String getName()
    {
        return name;
    }

    public String getNoise()
    {
        return noise;
    }

    public void verse()
    {
        System.out.println("And on that farm he had a " + name);
        System.out.println("ei ei o");
        System.out.println("With an " + noise + " " + noise + " here");
        System.out.println("And a " + noise + " " + noise + " there");
        System.out.println("Here a " + noise + " there a " + noise);
        System.out.println("Everywhere a " + noise + " " + noise);
    }
}

```

```
public class MacVerse
{
```

class definition header

```
    // animal name in verse
    private String name;
    // animal noise in verse
    private String noise;

    public MacVerse(String animalName, String animalNoise)
    {
        name = animalName;
        noise = animalNoise;
    }

    public String getName()
    {
        return name;
    }

    public String getNoise()
    {
        return noise;
    }

    public void verse()
    {
        System.out.println("And on that farm he had a " + name);
        System.out.println("ei ei o");
        System.out.println("With an " + noise + " " + noise + " here");
        System.out.println("And a " + noise + " " + noise + " there");
        System.out.println("Here a " + noise + " there a " + noise);
        System.out.println("Everywhere a " + noise + " " + noise);
    }
}
```

```

public class MacVerse
{
    // animal name in verse
    private String name;
    // animal noise in verse
    private String noise;

    public MacVerse(String animalName, String animalNoise)
    {
        name = animalName;
        noise = animalNoise;
    }

    public String getName()
    {
        return name;
    }

    public String getNoise()
    {
        return noise;
    }

    public void verse()
    {
        System.out.println("And on that farm he had a " + name);
        System.out.println("ei ei o");
        System.out.println("With an " + noise + " " + noise + " here");
        System.out.println("And a " + noise + " " + noise + " there");
        System.out.println("Here a " + noise + " there a " + noise);
        System.out.println("Everywhere a " + noise + " " + noise);
    }
}

```

```

public class MacVerse
{
    // animal name in verse
    private String name;
    // animal noise in verse
    private String noise;

    public MacVerse(String animalName, String animalNoise)
    {
        name = animalName;
        noise = animalNoise;
    }

    public String getName()
    {
        return name;
    }

    public String getNoise()
    {
        return noise;
    }

    public void verse()
    {
        System.out.println("And on that farm he had a " + name);
        System.out.println("ei ei o");
        System.out.println("With an " + noise + " " + noise + " here");
        System.out.println("And a " + noise + " " + noise + " there");
        System.out.println("Here a " + noise + " there a " + noise);
        System.out.println("Everywhere a " + noise + " " + noise);
    }
}

```

Attributes


```

public class MacVerse
{
    // animal name in verse
    private String name;
    // animal noise in verse
    private String noise;

    public MacVerse(String animalName, String animalNoise)
    {
        name = animalName;
        noise = animalNoise;
    }

    public String getName()
    {
        return name;
    }

    public String getNoise()
    {
        return noise;
    }

    public void verse()
    {
        System.out.println("And on that farm he had a " + name);
        System.out.println("ei ei o");
        System.out.println("With an " + noise + " " + noise + " here");
        System.out.println("And a " + noise + " " + noise + " there");
        System.out.println("Here a " + noise + " there a " + noise);
        System.out.println("Everywhere a " + noise + " " + noise);
    }
}

```

```

public class MacVerse
{
    // animal name in verse
    private String name;
    // animal noise in verse
    private String noise;

    public MacVerse(String animalName, String animalNoise)
    {
        name = animalName;
        noise = animalNoise;
    }

    public String getName()
    {
        return name;
    }

    public String getNoise()
    {
        return noise;
    }

    public void verse()
    {
        System.out.println("And on that farm he had a " + name);
        System.out.println("ei ei o");
        System.out.println("With an " + noise + " " + noise + " here");
        System.out.println("And a " + noise + " " + noise + " there");
        System.out.println("Here a " + noise + " there a " + noise);
        System.out.println("Everywhere a " + noise + " " + noise);
    }
}

```

Constructor

```
public class MacVerse
{
    // animal name in verse
    private String name;
    // animal noise in verse
    private String noise;

    public MacVerse(String animalName, String animalNoise)
    {
        name = animalName;
        noise = animalNoise;
    }
}
```

```
public String getName()
{
    return name;
}

public String getNoise()
{
    return noise;
}

public void verse()
{
    System.out.println("And on that farm he had a " + name);
    System.out.println("ei ei o");
    System.out.println("With an " + noise + " " + noise + " here");
    System.out.println("And a " + noise + " " + noise + " there");
    System.out.println("Here a " + noise + " there a " + noise);
    System.out.println("Everywhere a " + noise + " " + noise);
}
}
```

Additional
Methods

Back to the big picture

```

public class MacSong
{
    public static void main(String[] args)
    {
        MacChorus chorus = new MacChorus();
        MacVerse pig = new MacVerse("pig", "oink");
        MacVerse dog = new MacVerse("dog", "woof");
        chorus.showChorus();
        pig.verse();
        chorus.showChorus();
        chorus.showChorus();
        dog.verse();
        chorus.showChorus();
    }
}

```

```

public class MacVerse
{
    // animal name in verse
    private String name;
    // animal noise in verse
    private String noise;

    public MacVerse(String animalName, String animalNoise)
    {
        name = animalName;
        noise = animalNoise;
    }

    public String getName()
    {
        return name;
    }

    public String getNoise()
    {
        return noise;
    }

    public void verse()
    {
        System.out.println("And on that farm he had a " + name);
        System.out.println("ei ei o");
        System.out.println("With an " + noise + " " + noise + " here");
        System.out.println("And a " + noise + " " + noise + " there");
        System.out.println("Here a " + noise + " there a " + noise);
        System.out.println("Everywhere a " + noise + " " + noise);
    }
}

```

```

// Provides chorus for Old MacDonald
public class MacChorus
{
    public void showChorus()
    {
        System.out.println("Old Macdonald had a farm");
        System.out.println("ei ei o");
    }
}

```

Go to CodeTracer

Types

- Variables that you declare have to be of some *type*:
 - `int x;`
 - `double gasMileage;`
 - `Car myMaserati;`

Types

- Some are objects, and some are not.

- Primitive types

Type	Bits	Range	Comment
byte	8	-128 to 127	used when only small whole numbers are considered
short	16	-32,768 to 32,767	used for whole numbers
int	32	-2,147,483,648 to 2,147,483,647	common type for whole numbers
long	64	$\sim -9 \times 10^{18}$ to $\sim +9 \times 10^{18}$	also common; used when large integers are possible
float	32	$\sim -3 \times 10^{38}$ to $\sim +3 \times 10^{38}$	represents decimal numbers
double	64	$\sim -1.7 \times 10^{308}$ to $\sim +1.7 \times 10^{308}$	common representation for decimal numbers
char	16	represents characters	supports 65,536 distinct characters
boolean		true or false	especially important for testing role in control statements

- Primitive types

Type	Bits	Range	Comment
byte	8	-128 to 127	used when only small whole numbers are considered
short	16	-32,768 to 32,767	used for whole numbers
int	32	-2,147,483,648 to 2,147,483,647	common type for whole numbers
long	64	$\sim -9 \times 10^{18}$ to $\sim +9 \times 10^{18}$	also common; used when large integers are possible
float	32	$\sim -3 \times 10^{38}$ to $\sim +3 \times 10^{38}$	represents decimal numbers
double	64	$\sim -1.7 \times 10^{308}$ to $\sim +1.7 \times 10^{308}$	common representation for decimal numbers
char	16	represents characters	supports 65,536 distinct characters
boolean		true or false	especially important for testing role in control statements

booleans

- `boolean a;`
- `a=false;`
- `boolean b=true;`

- later in the course:
 - if (`a<3`)
 - do some stuff....

char vs. String

- "Erik" is a String with 4 letters.
- "E" is a String with 1 letter.
- 'E' is a char.
- 'Erik' is an error in Java.

More about Strings

- String class is an example of a class that was designed by someone else.
 - but you can use it!
 - We have already done some things, but there is much more.
 - A great way to learn about *methods!*

More about Strings

- `String s="Erik";`
- `s.length()`
4
- `s.charAt(0)`
'E'
- `s.charAt(3)`
'k'
-

More about Strings

- `s.charAt(4)`

```
java.lang.StringIndexOutOfBoundsException: String index out of range: 4 at java.lang.String.charAt (String.java:686)
```

More about Strings

- `s="Ping Pong";`
- `s.toUpperCase();`
PING PONG

IMPORTANT

- `s.toUpperCase()`
does not change s !!!!!
- In order to keep the results, we must do an *assignment*:
`upper_s = s.toUpperCase();`
or
`s = s.toUpperCase();`

DrJava

import

- To use code that other people have written
 - `import java.util.*;`
 - `import java.util.Date;`
 - `import java.util.Scanner;`

API

- Application Program Interface

Getting to the Java API



Class TreeHouse

```
java.lang.Object  
└─ TreeHouse
```

```
public class TreeHouse  
extends java.lang.Object
```

A class for making and characterizing tree houses

Field Summary

private int	heightAboveGround Tree house height above ground
private int	length Tree house length
private java.lang.String	treeKind Tree house kind of tree, e.g. oak
private int	width Tree house width

Constructor Summary

[TreeHouse](#)(int theWidth, int theLength, int theHeight, java.lang.String tree)
the tree house constructor

Method Summary

int	area () Gives tree house area
boolean	dangerous () Determines if tree house is dangerous - 10 or more feet high
int	getHeight ()

Class TreeHouse

```
java.lang.Object  
└─ TreeHouse
```

```
public class TreeHouse  
extends java.lang.Object
```

A class for making and characterizing tree houses

Field Summary

private int	heightAboveGround Tree house height above ground
private int	length Tree house length
private java.lang.String	treeKind Tree house kind of tree, e.g. oak
private int	width Tree house width

Constructor Summary

```
TreeHouse(int theWidth, int theLength, int theHeight, java.lang.String tree)  
the tree house constructor
```

Method Summary

int	area() Gives tree house area
boolean	dangerous() Determines if tree house is dangerous - 10 or more feet high
int	getHeight()



Java™ 2 Platform Standard Ed. 5.0

[All Classes](#)

Packages

[java.applet](#)

[java.awt](#)

[java.awt.color](#)

All Classes

[AbstractAction](#)

[AbstractBorder](#)

[AbstractButton](#)

[AbstractCellEditor](#)

[AbstractCollection](#)

[AbstractColorChooserP](#)

[AbstractDocument](#)

[AbstractDocument.Attri](#)

[AbstractDocument.Con](#)

[AbstractDocument.Elen](#)

[AbstractExecutorServic](#)

[AbstractInterruptibleCh](#)

[AbstractLayoutCache](#)

[AbstractLayoutCache.N](#)

[AbstractList](#)

[AbstractListModel](#)

[AbstractMap](#)

[AbstractMethodError](#)

[AbstractPreferences](#)

[AbstractQueue](#)

[AbstractQueuedSynchr](#)

[AbstractSelectableChar](#)

[AbstractSelectionKey](#)

Overview Package Class Use Tree Deprecated Index Help

PREV NEXT

FRAMES NO FRAMES

Java™ 2 Platform
Standard Ed. 5.0

Java™ 2 Platform Standard Edition 5.0 API Specification

This document is the API specification for the Java 2 Platform Standard Edition 5.0.

See:

[Description](#)

Java 2 Platform Packages

java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.

String class in Java API

Scanner class

```
1 import java.util.Scanner; // imports just the Scanner class from java.util
2
3 public class MacVerseTester
4 {
5     public static void main(String[] args)
6     {
7         Scanner scan = new Scanner(System.in); // creates a Scanner object
8         System.out.println("Enter an animal name"); // a prompt
9         String animal = scan.next(); // waits for your keyboard input
10        System.out.println("Enter that animal's noise");
11        String noise = scan.next();
12        MacVerse someAnimal = new MacVerse(animal,noise); // makes a MacVerse object
13        someAnimal.verse(); // prints the verse
14    }
15 }
```

Scanner class

```
import java.util.Scanner;

public class MyAdder
{
    public static void main (String[] args)
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter two decimal numbers");
        double num1 = scan.nextDouble();
        double num2 = scan.nextDouble();
        System.out.println("The sum of " + num1 + " " + num2 + " is ");
        System.out.println(num1 + num2);
    }
}
```

End for today