### **Object Oriented Paradigms**

College Requirements

**CSCR2105** 

# Classes

Lecture 2.2



Because strings are so common, we don't have to use the new operator to create a String object

```
title = "Java Software Solutions";
```

- This is special syntax that works <u>only</u> for strings objects
- Each string (enclosed in double quotes) represents a **String** object



- Once a String object has been created, neither its value nor its length can be changed
- Thus we say that **an object** of the **String** class is **immutable**
- However, several methods of the String class return new String objects that are modified versions of the original
- See the list of String methods on chapter 16
- You can also use the API documentation



- It is occasionally helpful to refer to a particular character within a string
- This can be done by specifying the character's numeric index
- The indexes begin at zero in each string
- In the string "Hello", the character
   'H' is at index 0 and the 'o' is at index

## Example



```
String phrase = new String ("Change is inevitable");
  String m1, m2, m3, m4;
  System.out.println ("Original string: \"" + phrase + "\"");
  System.out.println ("Length of string: " + phrase.length());
  m1= phrase.concat (", except from vending machines.");
  m2= m1.toUpperCase();
  m3= m2.replace ('e', 'x');
  m4 = m3.substring(3, 30);
  // Print each mutated string
  System.out.println ("Text #1: " + m1);
  System.out.println (" Text #2: " + m2);
  System.out.println (" Text #3: " + m3);
  System.out.println (" Text #4: " + m4);
  System.out.println (" Text length: " + m4.length());
```



- The Math class is part of the java.lang package
- The Math class contains methods that perform various mathematical functions
- These include:
  - o absolute value
  - square root
  - exponentiation
  - trigonometric functions



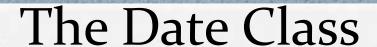
### The Math Class

- The methods of the Math class are static methods (also called class methods)
- Static methods can be invoked through the class name – no object of the Math class is needed

```
value = Math.cos(90) + Math.sqrt(delta);
```

See Chapter 6

```
int a, b, c; // ax^2 + bx + c
  double dis, root1, root2;
  Scanner scan = new Scanner (System.in);
System.out.print ("Enter the coefficient of x squared: ");
  a = scan.nextInt();
 System.out.print ("Enter the coefficient of x: ");
  b = scan.nextInt();
 System.out.print ("Enter the constant: ");
  c = scan.nextInt();
 dis= Math.pow(b, 2) - (4 * a * c);
  root1 = ((-1 * b) + Math.sqrt(dis)) / (2 * a);
  root2 = ((-1 * b) - Math.sqrt(dis)) / (2 * a);
```



Java provides a system-independent encapsulation of date and time in the <u>java.util.Date</u> class. You can use the <u>Date</u> class to create an instance for the <u>current</u> date and time and use its <u>toString</u> method to return the date and time as a string.

The + sign indicates public modifer +Date() +Date(elapseTime: long) +toString(): String +getTime(): long +setTime(elapseTime: long): void

Constructs a Date object for the current time.

Constructs a Date object for a given time in milliseconds elapsed since January 1, 1970, GMT.

Returns a string representing the date and time.

Returns the number of milliseconds since January 1, 1970, GMT.

Sets a new elapse time in the object.

## The Date Class Example

For example, the following code

```
java.util.Date date = new java.util.Date();
System.out.println(date.toString());
```

displays a string like Mon Nov 14 09:50:19 EST 2016.



### The Random Class



You have used <u>Math.random()</u> to obtain a random double value between 0.0 and 1.0 (excluding 1.0). A more useful random number generator is provided in

#### java.util.Random

+Random()

+Random(seed: long)

+nextInt(): int

+nextInt(n: int): int

+nextLong(): long

+nextDouble(): double

+nextFloat(): float

+nextBoolean(): boolean

Constructs a Random object with the current time as its seed.

Constructs a Random object with a specified seed.

Returns a random int value.

Returns a random int value between 0 and n (exclusive).

Returns a random long value.

Returns a random double value between 0.0 and 1.0 (exclusive).

Returns a random float value between 0.0F and 1.0F (exclusive).

Returns a random boolean value.



If two <u>Random</u> objects have the same seed, they will generate identical sequences of numbers. For example, the following code creates two <u>Random</u> objects with the same seed 3.

```
Random random1 = new Random(3);
System.out.print("From random1: ");
for (int i = 0; i < 10; i++)
   System.out.print(random1.nextInt(1000)+" ");
Random random2 = new Random(3);
System.out.print("\nFrom random2: ");
for (int i = 0; i < 10; i++)
   System.out.print(random2.nextInt(1000)+" ");</pre>
```

From random1: 734 660 210 581 128 202 549 564 459 961 From random2: 734 660 210 581 128 202 549 564 459 961



- A class library is a collection of classes that we can use when developing programs
- The Java standard class library is part of any Java development environment
- Its classes are not part of the Java language perose, but we rely on them heavily
- Various classes we've already used (System, Scanner, String) are part of the Java standard class library
- Other class libraries can be obtained through third party vendors, or you can create them yourself



- The classes of the Java standard class library are organized into packages
- Some of the packages in the standard class library are:

#### **Package**

#### **Purpose**

java.lang

java.applet

java.awt

javax.swing

java.net

java.util

General support

Creating applets for the web

Graphics and graphical user interfaces

Additional graphics capabilities

**Network communication** 

**Utilities** 

javax.xml.parsers XML document processing



When you want to use a class from a package, you could use its fully qualified name

```
java.util.Scanner;
java.util.Random;
```

Or you can import the class, and then use just the class name

```
import java.util.Scanner;
```

To import all classes in a particular package, you can use the \* wildcard character

```
import java.util.*;
```



- All classes of the java.lang package are imported automatically into all programs
- It's as if all programs contain the following line: import java.lang.\*;
- That's why we didn't have to import the System or String classes explicitly in earlier programs
- The Scanner class, on the other hand, is part of the java.util package, and therefore must be imported

# Summary

- String
- Math
- Date
- Random

## **NOW:**

Waiting for your questions and comments

Object-Oriented Programming: Inheritance