

Mat 2170
Week 12

**Strings,
Classes, and
Formatting**

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

Mat 2170 Week 12

Strings, Classes, and Formatting

Spring 2014

Student Responsibilities

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

- Reading: Textbook, Review sections:
 - 5.1 – Methods overview
 - 5.2 – Writing methods
 - 5.3 – Mechanics of method calls
 - 6.3 – Defining your own classes
 - 6.4, 6.5 – Class examples
 - 8.2 – Characters
 - 8.4 – String methods
- Lab: Character and String processing;
Writing a class from scratch
- Attendance

Commonly Used String Methods

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

- To select a character from a string: `charAt(ndx)`

```
char firstChar = str.charAt(0);
```

- To Concatenate two strings: use `+`

```
String strC = strA + strB;
```

- To extract a substring:
`substring(startPosn, OnePastEndPosn)`

```
String newStr = str.substring(p1, p2);
```

where `p1` is the first index position in the desired substring, and `p2` is the index position in the target string **immediately following the last position** of the substring.

- To check equality of two strings: `equals(str2)`

```
if (s1.equals(s2))...
```

or

```
if (s1.equalsIgnoreCase(s2))...
```

- To obtain a **copy** of a string with the case of letters in it changed:

```
s1.toUpperCase()
```

or

```
s1.toLowerCase()
```

Recall: These methods **do not change** the receiver.

For that, you need: `s1 = s1.toUpperCase()`

- To search in a string:

<code>indexOf(ch)</code>	<code>indexOf(str)</code>
<code>indexOf(ch, startPosn)</code>	<code>indexOf(str, startPosn)</code>

Example: str											
0	1	2	3	4	5	6	7	8	9	10	11
h	e	l	l	o	,		w	o	r	l	d

```
str.indexOf('h')           returns 0
```

```
str.indexOf("o")          returns 4
```

```
str.indexOf("ell")        returns 1
```

```
str.indexOf('x')          returns -1
```

```
str.indexOf("o", 5)       returns 8
```

Two Important String Idioms

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

1. Iterating through the characters in a string:

```
for (int i = 0; i < str.length(); i++)  
{  
    char ch = str.charAt(i);  
    ...code to process each character in turn...  
}
```

2. Growing a new string character by character:

```
String result = "";  
for (whatever limits are appropriate)  
{  
    ...code to determine next char to be added...  
    result += ch;  
}
```

Revisiting Palindromes

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

- Recall that we recently wrote a numeric palindrome program.
- The first exercise in this week's lab is to write a program to test strings to see if they are palindromes.
- This will be a **yes it is** or **no it isn't** decision; do **not** attempt to force a string into a palindrome as we did in the integers.
- Use the input method `readLine()` to get a line of input from the user as a `String` — provide a user prompt in the same way you did with `readInt()` and `readDouble()`.

- If we can make a **string** and its **reverse uniform**, then testing for equality will be sufficient to determine whether a string is a palindrome.
- Make this program **modular** — design and utilize meaningful methods logically. To help, create and utilize the methods:
 1. **cleanString()** to remove all non-alphabetic characters
 2. **Reverse()** to obtain the reverse of a string
 3. **isSentencePalindrome()** to determine whether the string is a palindrome or not.
- Your program is to be capable of processing several lines of input — and should halt when the input string is empty.

Formatting Decimal Output

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

There is a java **formatting** class which allows us to specify the number of digits following the decimal point when displaying double values.

- **Import** the necessary library:

```
import java.text.DecimalFormat;
```

- **Instantiate** a formatting object (the one below insists on a zero before the decimal point if the number is less than one, and displays two digits after the decimal point):

```
DecimalFormat patternDot2 =  
    new DecimalFormat("####0.00");
```

- **Use** the formatting object:

```
println (patternDot2.format(  
    A10.getMeasureInRadians()) + " Radians \n");
```

Writing Classes from “Scratch”

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

- The second exercise in lab this week is to create a class from “scratch” — i.e., it won't be explicitly extended from another class.
- You'll be creating:
 1. a **Temperature** project within a `lab12` directory
 2. Inside the **Temperature** project, create a **Temperature** package
 3. which in turn will house your **Temperature** class.

`lab12` ⇒ `Temperature Project` ⇒ `Package` ⇒ `Class`

- The test program, `TestTemperature.java`, is available on our web site, and should be downloaded into `lab12/Temperature/src` — into the default package.
- **Do not modify** the `run()` method in the test program other than to move the `/*` down as methods are completed.
- Adhere to the naming conventions required in the lab write-up.

The Temperature Class

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

- Classes consist of **data members** and **member methods**: be sure you understand what members are required, what their names should be, and the return types of member methods.
- Data members:
 1. a floating-point value
 2. a character for the scale: 'C' for Celsius or 'F' for Fahrenheit
 3. a numeric formatter.
- Lab 12 write-up indicates the methods required.
- When you have the class implemented correctly, the output should match that on the next slide.

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review


Palindromes

Formatting

Temperature

Rational Class

Message

 The constructor TEST results are:
Default and double (zero) -- OKAY
Default and 'C' (zero) -- OKAY
Default and two-param (zero) -- OKAY
Bad scale replaced with 'C', one-param -- OKAY
Bad scale replaced with 'C', two-param -- OKAY

TEST equivalent temperatures:
Same at -40.0 -- OKAY
Same at freezing -- OKAY
Same at boiling -- OKAY

TEST toString() and getDegreesC()/getDegreesF():
toString(): 100.00 Degrees C --- in F is: 212.00
toString(): 212.00 Degrees F --- in C is: 100.00

TEST inequalities
isLess() -- OKAY
isGreater() -- OKAY

TEST mutators
Should be 98.60F: 98.60 Degrees F
Should be 0.00F: 0.00 Degrees F
Should be 98.60F: 98.60 Degrees F
Should be 212.0C: 212.00 Degrees C

TEST add()
Should be 312.0C: 312.00 Degrees C

last TEST
Display 123.4567 F, rounded to 2 decimal places:
123.46 Degrees F, and equiv: 50.81 Celsius

END of TESTING

OK

Rational Numbers — Creating a Class

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

In Mathematics:

$$Q = \left\{ \frac{a}{b} \mid a, b \in \text{Integers} \wedge b \neq 0 \right\}$$

The Rules of Rational Numbers — **Invariants**

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

1. The denominator should always be positive.
2. The fraction should always be reduced to lowest terms.
3. The `Rational` class should be immutable — once a `Rational` object is constructed, the client should not be allowed to modify it.

Rational Class Data Members

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

- A Rational object needs to store an integer **numerator** and **denominator**.
- The class must ensure the numerator and denominator are **reduced to lowest terms** (divide by the greatest common divisor)
- **Access should be restricted** to the class itself — clients should not be able to manipulate the data members.

Rational Class Member Methods

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

■ **Constructors:**

- `Rational(n, d)` — two parameter, $\frac{n}{d}$ in lowest terms
- `Rational(n)` — single parameter, $\frac{n}{1}$
- `Rational()` — default (no parameter), $\frac{0}{1}$
- `Rational(F)` — copy another Rational object

■ **Inspectors:** `getNumerator()`, `getDenominator()`

■ **Mutators** — not allowed for this class

■ **Facilitators:**

`gcd()`, `reduce()`, `toString()`,
`add()`, `subtract()`, `multiply()`, `divide()`

Euclid's gcd() Algorithm

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

```
// Facilitators
// Greatest Common Divisor method
//   (for use in constructor)

private int gcd(int x, int y) {
    int r = x % y;
    while (r != 0) {
        x = y;
        y = r;
        r = x % y;
    }
    return y;
}
```

Mat 2170
Week 12

**Strings,
Classes, and
Formatting**

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

Create and test the Rational class

Calculating π — An Approximation

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

$$\pi = 2 \left(1 + \frac{1}{3} + \frac{1 \cdot 2}{3 \cdot 5} + \frac{1 \cdot 2 \cdot 3}{3 \cdot 5 \cdot 7} + \dots \right)$$

- We can approximate π using the first n terms of the summation.
- For example, if $n = 2$, the result is $2(1 + \frac{1}{3}) = \frac{8}{3}$.

Calculating π — Methods

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

- `piApproximation()` — a public method which will take a single integer, n , and return a `Rational` object.
- `toDouble()` — a public method which yields a floating point representation of a `Rational` object — to view the approximation of π as a decimal number.

Formatting Decimal Output

Mat 2170
Week 12

Strings,
Classes, and
Formatting

Week 12

String Review

Palindromes

Formatting

Temperature

Rational Class

```
■ import java.text.DecimalFormat;

■ DecimalFormat patternDot2 =
    new DecimalFormat("####0.00");

■ println (patternDot2.format(
    A10.getMeasureInRadians()) + " Radians \n");
```