COMP 110-003
Introduction to Programming
Review Strings and Loops

February 21, 2013

Haohan Li
TR 11:00 – 12:15, SN 011
Spring 2013

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

indexOf() and substring()

2	.	5		+		3
0	1	2	3	4	5	6

- `String s = "2.5 + 3";`
- `int p1 = s.indexOf(" ");`
- `int p2 = s.lastIndexOf(" ");`
- `System.out.println(p1 + "," + p2);`
- The output will be 3,5

indexOf() and substring()

2	.	5		+		3
0	1	2	3	4	5	6

- `String s = "2.5 + 3";`
- `int p1 = s.indexOf(" ");`
- `int p2 = s.lastIndexOf(" ");`
- `System.out.println(p1 + "," + p2);`
- The output will be **3,5**

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

indexOf() and substring()

2	.	5		+		3
0	1	2	3	4	5	6

- `String operand1 = s.substring(0, p1);`

indexOf() and substring()

2	.	5		+		3
0	1	2	3	4	5	6

- `String operand1 = s.substring(0, p1);`
- `String operator = s.substring(p1, p2);`

indexOf() and substring()

2	.	5		+		3
0	1	2	3	4	5	6

- `String operand1 = s.substring(0, p1);`
- `String operator = s.substring(p1, p2);`
- `String operand2 = s.substring(p2);`

indexOf() and substring()

2	.	5		+		3
0	1	2	3	4	5	6

- `String operand1 = s.substring(0, p1);`
- `String operator = s.substring(p1, p2);`
- `String operand2 = s.substring(p2);`
- `System.out.println("***" + operand1 + "***" + operator + "***" + operand2 + "***");`
- The output will be `***2.5*** +*** 3***`

indexOf() and substring()

2	.	5		+		3
0	1	2	3	4	5	6

- `String operator = s.substring(p1, p2);`
- `System.out.print("***" + operator.equals("+") + "***");`
- `System.out.println("***" + operator.equals(" +") + "***");`
- The output will be **`***false******true***`**

indexOf() and substring()

2	.	5		+		3
0	1	2	3	4	5	6

- `String operand1 = s.substring(0, p1);`
- `String operator = s.substring(p1+1, p2);`
- `char oper = s.charAt(p1 + 1);`
- `String operand2 = s.substring(p2+1);`

indexOf() and substring()

2	.	5		+		3
0	1	2	3	4	5	6

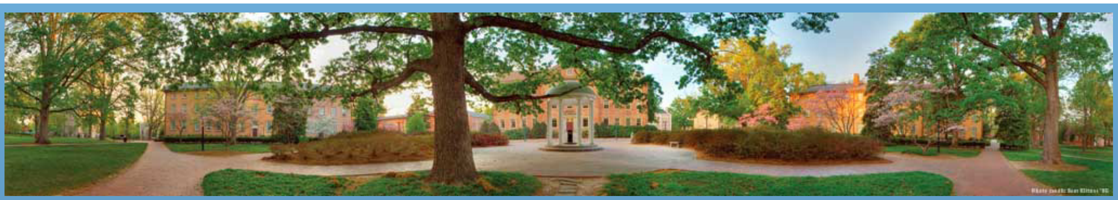
+		3
0	1	2

```
String newString = s.substring(p1 + 1);
System.out.println("***" + newString + "***");
int newP = newString.indexOf(" ");
String newString2 = newString.substring(newP + 1);
System.out.println("***" + newString2 + "***");
```

Extract Words

```
String t = "2.5 + 3 + 5 + 12 + 16";
while (t.indexOf(" ") != -1) {
    String temp = t.substring(0, t.indexOf(" "));
    t = t.substring(t.indexOf(" ") + 1);
    System.out.print("**" + temp + "**");
}
System.out.println("**" + t + "**");
```

- This piece of code will extract each single word in the string
  - While there is at least one space, we print the first word
- The output will be ****2.5**+**3**+**5**+**12**+**16****



Problem Solving Skills

- In our syllabus:
  - (1) Fundamental computer programming skills
  - (2) **Systematic and logical thinking**

Divide and Conquer

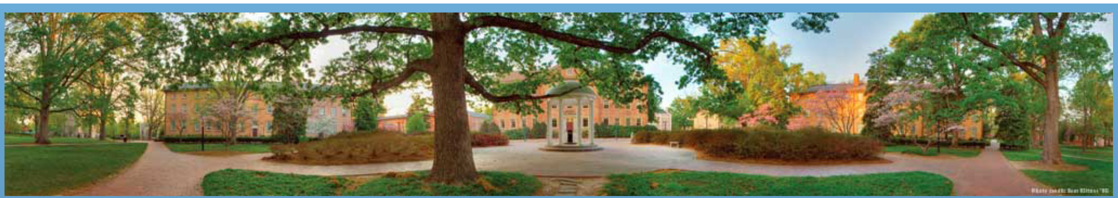
- **Breaking down a large, complex problem into smaller, solvable problems**
- In program 2, there should be 3 small problems
  - Partition the string into 3 substrings
  - Parse the substrings into float and char
  - Calculate the results

Reduction

- **Transforming the problem into another problem for which solutions exist**
- In program 2, if you know how to generate a substring with given indices, you should think about "can I find a way to generate these indices"?
- Then you read Figure 2.5 and find indexOf () method. Now since you know how to find an index, the extraction of word follows.

Reduction

- **Transforming the problem into another problem for which solutions exist**
- In program 2, if you know how to extract remove the first word of a string, can you extract all words?
- Of course you can! You can extract and remove one word. Then repeat the same procedure on the string with the first word removed – it is the old problem again and we know how to solve it!



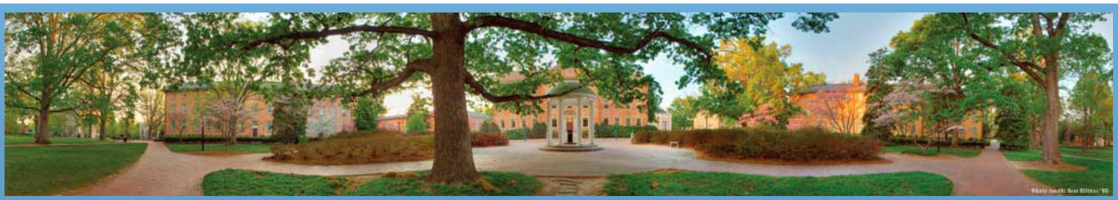
Trial-and-error

- **Testing possible solutions until the right one is found**
- In program 2, if you are not sure how to extract the exact operator
- You can try all these statements, output every result, and find the correct one
  - `String operator = s.substring(p1, p2);`
  - `String operator = s.substring(p1, p2+1);`
  - `String operator = s.substring(p1+1, p2);`

Root Cause Analysis

- **Identifying the cause of a problem**
- If you write the following code

  -
    ```
    if (operator.equals("+")) {
    // do something
    }
    ```
- You can let operator = "+", then run the code again
- Now it is working. You should guess the reason is that the value of operator is not "+"
- The point is: narrowing down the position of the bug



General Rules in Programming

- Divide a program into small parts. When you finish a part, make sure it is correct
- Print variables to check if they are in the same values as expected
- When you find a bug, make sure that your input is correct. Then run the program using only this input and print all temporary variables. If you can find the first statement when something goes wrong, you are very close to fix it

Power Operator

```
float foperand1 = Float.parseFloat(operand1);
float foperand2 = Float.parseFloat(operand2);
double result;
switch (oper) {
    case '^':
    result = 1.0;
    if (foperand2 > 0) {
        for (int count = 0; count < (int) foperand2; count++)
            result *= foperand1;
    } else if (foperand2 < 0) {
        for (int count = 0; count > (int) foperand2; count--) {
            result *= foperand1;
        }
        result = 1.0 / result;
    }
    break;
}
```