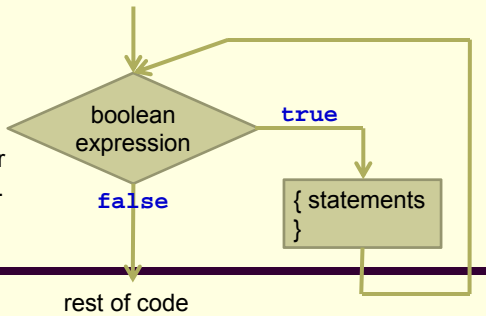


CS 259

Java's **while** Loop

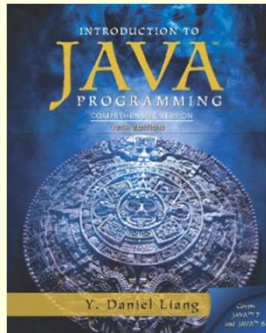
Instructor:
Joel Castellanos
e-mail: joel@unm.edu
Web: <http://cs.unm.edu/~joel/>
Office: Electrical and Computer
Engineering building (ECE).
Room 233



9/9/2016

Textbook & Reading Assignment

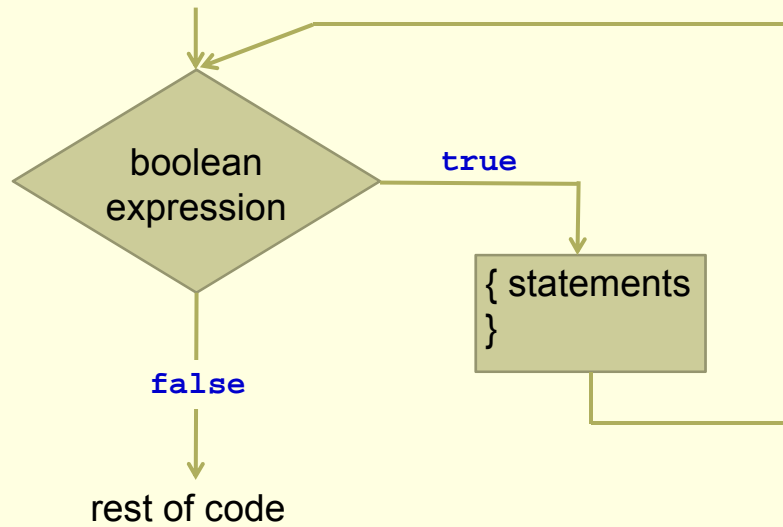
Introduction to Java Programming (10th Edition) by Y. Daniel Liang



Read by Monday: Sept 5
Chapter 5: Loops

Read by Wednesday: Sept 14
Chapter 6: Methods

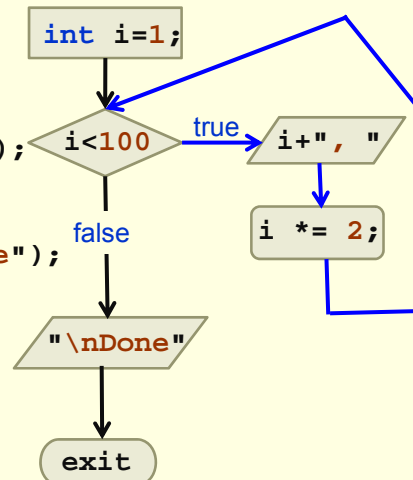
Java's while Loop



3

While Loop and Flow Chart

```
public class HelloWorld
{ public static void main(String[] args)
{
    int i = 1;
    while (i < 100)
    {
        System.out.print(i+", ");
        i *= 2;
    }
    System.out.println("\nDone");
}
}
```



Output:

```
1, 2, 4, 8, 16, 32, 64,
Done
```

4

While Loop: continue and break

```
1) int x = 1;
2) while (x < 10)
3) { x++;
4)   System.out.print("x="+x+" ");
5)   if (x % 3 == 0)
6)     { System.out.println("DO ITTA");
7)       continue;
8)     }
9)   else if (x % 7 == 0)
10)    { System.out.println("DOKE");
11)      break;
12)    }
13) System.out.println("OHA");
14)}
```

Since there is a `continue` statement on line 7, line 9 can be changed from `else if` to `if` without changing the program's behavior.

Output:

x=2	OHA
x=3	DO ITTA
x=4	OHA
x=5	OHA
x=6	DO ITTA
x=7	DOKE

5

Quiz (while): What is the Output?

```
1) public class WhileLoopExample
2) {
3)   public static void main(String[] args)
4)   {
5)     int n = 1;
6)     while (n < 10)
7)     {
8)       System.out.print(n + ", ");
9)       n *= 2; //n=n*2;
10)    }
11)  }
12)}
```

- a) 1,
- b) 1, 2, 3, 4, 5, 6, 7, 8, 9,
- c) 1, 2, 4, 8,
- d) 1, 2, 4, 6, 8,
- e) 1, 2, 4,

6

Quiz: While Loop

```
int i = 5;
while (i < 20)
{ System.out.print(i+", ");
  i += 3;
}
```

What would be the Output of the above Java code?

- a) 5, 8, 11, 14, 17,
- b) 5, 8, 11, 14, 17, 20,
- c) 5, 15,
- d) 5, 9, 12, 15, 18,
- e) 5, 9, 12, 15, 18, 21,

7

Example of "Loop until good"

```
1) import javax.swing.JOptionPane;
2) public class Tmp
3) { public static void main(String[] args)
4) {
5)     String msg = null;
6)     while (true)
7)     {
8)         msg = JOptionPane.showInputDialog(null,
9)             "Enter number from 0 through 9");
10)
11)         if (msg == null) continue;
12)         if (msg.length() != 1) continue;
13)         if (!Character.isDigit(msg.charAt(0))) continue;
14)         break;
15)     }
16)
17)     int n = Integer.parseInt(msg);
18)     System.out.println("n="+n);
19) }
8 20) }
```

The order of these `if` statements is important:

- `msg.length()` will fail if `msg` is `null`.
- `msg.charAt(0)` will fail if `msg` is an empty String, "".

Quiz (**while**): What is the Output?

```
1) public static void main(String[] args)
2) {
3)     int n = 10;
4)     int z = n-1;
5)     while (z > 1)
6)     {
7)         if ((n % z) == 0)
8)         {
9)             System.out.print(z +", ");
10)        }
11)        z--;
12)    }
13)}
```

- a) 9, 8, 7, 6, 5, 4, 3, 2, 1,
- b) 9, 8, 7, 6, 5, 4, 3, 2,
- c) 8, 6, 4, 2,
- d) 5, 2,
- e) 5,

9

Problem: Extract the Number

Write a Java program that will:

- 1) Search the String

"Euler was born in 1707 in Switzerland."

- 2) Find the characters that are digits.
- 3) Convert String digits to a number.
- 4) Print the result.

10

Extract the Number: Part 1 of 2

```
1) public static void main(String[] args)
2) {
3)     String foo = "Euler was "
4)         + "born in 1707 in Switzerland.";
5)
6)     int start = 0;
7)     char c = foo.charAt(start);
8)
9)     while (Character.isDigit(c)==false)
10)    {
11)        start++;
12)        c = foo.charAt(start);
13)    }
14)    System.out.println("start="+ start);
```

Find index of 1st digit

start=18

11

Extract the Number: Part 2 of 4

```
1)     int end = start + 1;
2)     c = foo.charAt(end);
3)     while (Character.isDigit(c))
4)     { end++;
5)       c = foo.charAt(end);
6)     }
7)
8)
9)     String a = foo.substring(start, end);
10)    int b = Integer.parseInt(a);
11)
12)    System.out.println(a + " is a String");
13)    System.out.println(b + " is an int");
14)    System.out.println(foo.charAt(end-1));
15) }
```

Find index after last digit

Output:
1707 is a String
1707 is an int
7

12

Extract the Number: Find the Bugs!!!

```
public static void main(String[] args)
{
    String foo = "Euler was born in 1707 in Switzerland.";
    int start = 0;
    char c = foo.charAt(start);
    while (!Character.isDigit(c))
    {
        start++;
        c = foo.charAt(start);
    }

    int end = start+1;
    c = foo.charAt(end);
    while (Character.isDigit(c))
    {
        end++;
        c = foo.charAt(end);
    }

    String a = foo.substring(start, end);
    int x = Integer.parseInt(a);
    System.out.println(start + "->" + end + ", x=" + x);
}
13 }
```

Will This work for other strings?

- "Euler 17 born"
- "Euler 178923 born"
- "171 Euler Born"
- "7 Euler"
- What other cases should be checked?

Add print statements inside loop to show variables that change.

Bug Fix: Number at End of String

```
1) String foo = "Euler 1707";
2) int start = 0;
3) char c = foo.charAt(start);
4) while (!Character.isDigit(c))
5) { start++;
6)   c = foo.charAt(start);
7) }
8)
9) int end = start+1;
10) c = foo.charAt(end);
11) while (Character.isDigit(c))
12) {
13)   end++;
14)   c = foo.charAt(end);
15)   c = foo.charAt(end);
16) }
```

With the if statement added on line 14, "Euler 1707" now works.

But... there is another bug.....

Bug fix: `if (end >= foo.length()) break;`

`java.lang.StringIndexOutOfBoundsException:
String index out of range: 10`

14

Bug Fix: Single Digit Number at End of String

```
1) String foo = "Euler 7";
2) int start = 0;
3) char c = foo.charAt(start);
4) while (!Character.isDigit(c))
5) { start++;
6)   c = foo.charAt(start);
7) }
8)
9) int end = start+1;
10) c = foo.charAt(end);
```

Fix: Change to:
`int end = start;`

`java.lang.StringIndexOutOfBoundsException:
String index out of range: 7`

15

But... there is another bug.....

Bug Fix: String Does Not Contain a Number

```
1) String foo = "Euler";
2) int start = 0;
3) char c = foo.charAt(start);
4) while (!Character.isDigit(c))
5) { start++;
6)   c = foo.charAt(start);
7) }
```

`java.lang.StringIndexOutOfBoundsException:
String index out of range: 5`

```
if(start == foo.length())
{ System.out.println("No digits found");
  System.exit(0);
}
```

16

But... there is another bug.....

Bug Fix: Number Too Large for `int`

```
1) String foo = "Euler 12345678901 Switzerland";  
   ↓  
19) while (Character.isDigit(c))  
20) { end++;  
21)   if (end >= foo.length()) break;  
22)   ← if (end-start > 9) break;  
23)   c = foo.charAt(end);  
24) }  
25) }  
26) }  
27) String a = foo.substring(start, end);  
28) int x = Integer.parseInt(a);  
29) System.out.println(x);
```

Is this the correct handling?

java.lang.NumberFormatException:
For input string: "12345678901"

17

Test Cases

```
"Euler was born in 1707 in Switzerland."  
"Euler 17 born"  
"Euler 178923 born"  
"171 Euler Born"  
"7 Euler"  
"Born in 1707"  
"Born 7"  
"Two Numbers: 123 456"  
"Euler 12345678901 Switzerland."
```

Any others?

18

```

1) String foo = "Euler was born in 1707 in Switzerland.";
2)
3) int start = 0;
4) char c = foo.charAt(start);
5) while (!Character.isDigit(c))
6) { start++;
7)   if(start == foo.length())
8)   { System.out.println("No digits found");
9)     System.exit(0);
10)  }
11)  c = foo.charAt(start);
12) }
13)
14) int end = start;
15) c = foo.charAt(end);
16)
17) while (Character.isDigit(c))
18) { end++;
19)   if (end >= foo.length()) break;
20)   if (end-start > 9) break;
21)   c = foo.charAt(end);
22) }
23)
24) String a = foo.substring(start, end);
25) int x = Integer.parseInt(a);
26) System.out.println(x);

```

Extract the Number
(With all bug fixes)

19

Quiz: What is the Output

```

1) String foo = "6789";
2) int end = 0;
3) char c = foo.charAt(end);
4)
5) while (Character.isDigit(c))
6) { end++;
7)   if (end >= foo.length()) break;
8)   if (end > 7) break;
9)   c = foo.charAt(end);
10)  System.out.print(c);
11) }

```

a) 7 b) 9 c) 678
d) 789 e) 6789

20

Quiz: What is the Output

```
1) String boo = "ABC123";
2) int end = 3, n = 0;
3) char c = boo.charAt(end);
4)
5) while (Character.isDigit(c))
6) { end++;
7)   if (end >= boo.length()) break;
8)   c = boo.charAt(end);
9)   n = n + Character.getNumericValue(c);
10) }
11) System.out.println(n);
```

- | | | |
|-----------|--------|-------|
| a) ABC123 | b) 123 | c) 12 |
| d) 6 | e) 5 | |

21

Count Upper, Lower, Digits and Other: Program Structure

```
1) import javax.swing.JOptionPane;
2)
3) public class CountCharTypes
4) {
5)   public static void main(String[] args)
6)   {
7)     String str = JOptionPane.showInputDialog(
8)       null, "Enter String");
9)
10)    // 1) Initialize Counters.
11)
12)
13)
14)
15)
16)
17)
18)
19)
20)    // 2) Loop through str and count char types.
21)
22)
23)
24)
25)
26)
27)
28)
29)
30)
31)
32)
33)
34)
35)
36)
37)
38)
39)
40)    // 3) Output.
41)
42)
43)
44)
45)
46)
47)
48)
49)
50)
51) }
52)
53) }
```

22

Count Upper, Lower, Digits and Other: Part 1

```
10) // 1) Initialize Counters.
11) int upper = 0;
12) int lower = 0;
13) int digit = 0;
14) int other = 0;
15)
16) int i = 0; //loop variable. Index into str.
```

In computer science, a *counter*, is usually an integer variable that starts with a value of 0 and is incremented as needed to count something.

23

Count Upper, Lower, Digits and Other: Part 2

```
20) // 2) Loop through str and count char types.
21) while (i<str.length())
22) {
23)     char c = str.charAt(i);
24)
25)     //Four mutually exclusive options.
26)     if (Character.isUpperCase(c)) upper++;
27)     else if (Character.isLowerCase(c))
28)     { lower++;
29)     }
30)     else if (Character.isDigit(c)) digit++;
31)     else other++;
32)
33)     i++; //Increment on every iteration.
24 34) }
```

Count Upper, Lower, Digits and Other: Part 3

```
40) // 3) Output.
41) System.out.println("Input: " + str);
42) System.out.println("Uppercase Count = "
43)     + upper);
44)
45) System.out.println("Lowercase Count = "
46)     + lower);
47)
48) System.out.println("Digit Count = "
49)     + digit);
50) System.out.println("Other Count = "
51)     + other);
52) } //ends main
53) } //ends class
```

25

Doubly Nested Loop: What is the Output?

```
1) int i = 0;
2) while (i<5)
3) { int k = 1;
4)   while (k < 14)
5)     { System.out.print(k + ", ");
6)       k += 2;
7)     }
8)
9)   System.out.println("<==== " + i);
10)  i++;
11) }
```

```
1, 3, 5, 7, 9, 11, 13, <==== 0
1, 3, 5, 7, 9, 11, 13, <==== 1
1, 3, 5, 7, 9, 11, 13, <==== 2
1, 3, 5, 7, 9, 11, 13, <==== 3
1, 3, 5, 7, 9, 11, 13, <==== 4
```

26

Doubly Nested Loop: What is the Output?

```
1) int i = 0;
2) while (i<5)
3) { int k = i; //changed from: k=1;
4)   while (k < 14)
5)     { System.out.print(k + ", ");
6)       k += 2;
7)     }
8)
9)   System.out.println("<==== " + i);
10)  i++;
11) }
```

```
0, 2, 4, 6, 8, 10, 12, <==== 0
1, 3, 5, 7, 9, 11, 13, <==== 1
2, 4, 6, 8, 10, 12, <==== 2
3, 5, 7, 9, 11, 13, <==== 3
4, 6, 8, 10, 12, <==== 4
```

27

Quiz: Doubly Nested Loop

```
1) int n = 1;
2) while (n<8)
3) { int p = n;
4)   while (
5)     { System.out.print(p+"\t");
6)       p += n;
7)     }
8)   System.out.println();
9)   n++;
10) }
```

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16
5	10	15	20
6	12	18	24
7	14	21	28

Which logical expression placed in line 4 will result in the output shown in the table above?

- a) $p < n + 4$ b) $p < 5*n$
c) $n < n*n$ d) $n < n + 4$
e) $n > n + 4$

28

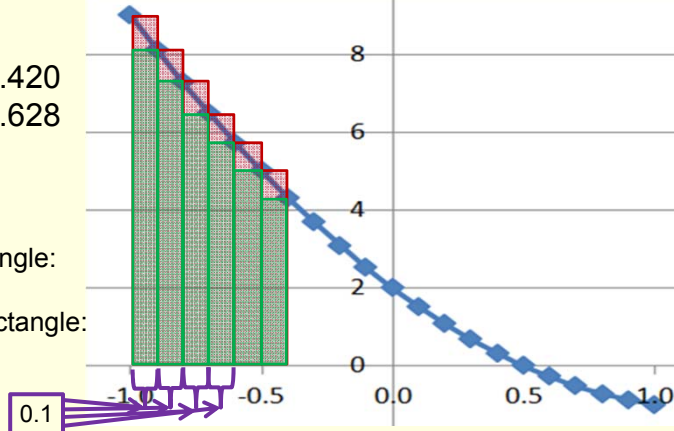
Lab 3: Area Under Curve

- Use a loop to have x range from [-1, 1].
- `double deltaX = 0.1;`
- `double y = Math.abs(2.0*x*x - 5.0*x + 2.0);`

Left Sum = 6.420
Right Sum = 5.628

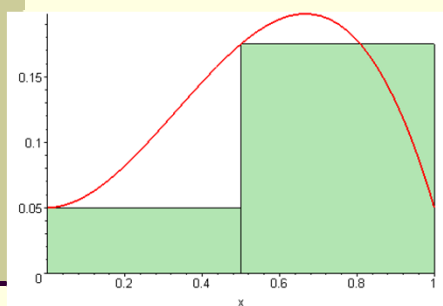
Area of *first red* rectangle:
= $y(-1.0) * 0.1$
Area of *first green* rectangle:
= $y(-1.1) * 0.1$

29

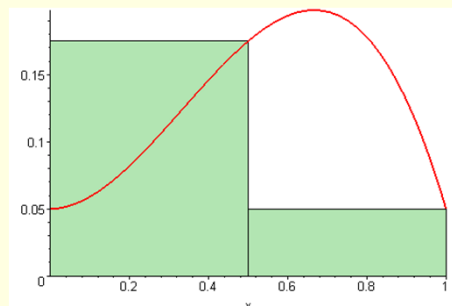


Riemann Sum Approximation of Area

- https://en.wikipedia.org/wiki/Riemann_sum



Left Sum



Right Sum

30

Lab 3 Requirements

- Write a program that uses a nested loop to calculate and display both the left and right sum for the following values of deltaX:

0.1, 0.01, 0.001, 0.0001, 0.00001, and 0.000001.

Hints: The variable you use for the total area must be set to zero **inside** the loop that decreases deltaX and **outside** the loop that sums the rectangles.

The **same** inner loop can be used to calculate both the left and right sum.

31

Find Prime Factors

100

■ / 2 = 50

■ / 2 = 25

■ / 2 no:

■ / 3 no:

■ / 4 no:

■ / 5 = 5

■ / 5 = 1

32