

Loops

Lecture Outline & Notes

Lecture Outline

1. Introduction
 - Processing data means performing actions on that data
 - Often, the same action needs to be performed on each element in a collection
 - Some actions may need to be repeated until a certain condition applies
 - In general: loops involve some *initialization*, termination condition, and an update toward the terminating condition
2. While Loop
 - a. Syntax:

```
while(<condition>) {  
  <repeated code block>  
}
```
 - b. Behavior: condition is checked before the loop's (entire) code block is executed; checked again before each iteration
 - c. Counter-controlled loop: a counter (index) is initialized prior to the while loop; progress is made inside the while loop to the terminating condition
 - d. Exercises:
 - Write code to compute the sum of 1..100
 - Write code to compute the product of 1..100
3. Do-while loop
 - a. Syntax:

```
do {  
  <repeated code block>  
} while (<condition>);
```
 - b. Behavior: the first iteration of the code block is executed *unconditionally*; terminating condition is tested at the *end* of a loop (note the semicolon)
 - c. F
4. For-loop
 - a. Syntax:

```
for(<init>; <terminating condition>; <update>) { ... }
```
 - b. All three constructs are in the same line/place; more readable
 - c. Examples
 - `for(int i=0; i<10; i++)`
 - C: ANSI C does not allow you to declare an index inside a for-loop's control structure; Java does

- Sum of 1..100
- Print all even numbers 1..100

5. Misc

- Increment/decrement operators
- Compound assignment operators
- Flag-controlled loop
- Sentinel controlled loop
 - Conditions in a loop are statements (need not be simple expressions): could use any statements (function calls) that return a usable (boolean or integer) value
 - Could even be empty if not needed:
`for(; i<n;) {...`
 - Example (C):
`printf("Please select an option (1-6:");
for(scanf("%d", &option); (option > 6 || option < 1); scanf("%d", &option)) { ..}`
- Nesting
 - Two dimensional data
 - The inner loop is executed *for each* iteration of an outer-for loop
- Enhanced-For loop (Java)
 - Java provides a convenience syntax for collections (arrays or anything that implements the Iterable interface):
`for(Type t : collection) { do something with t }`

6. Common Errors

- Syntax (semicolons, brackets)
- Proper initialization, most appropriate loop
- Off-by-one errors
- Infinite loops
 - Avoid floating point numbers
 - Zune bug

7. Exercises

- Write code snippets to produce the following patterns of numbers:
 - A list of even integers 0 to n, one to a line
 - The same list, but delimited by commas
 - A list of integers divisible by 3 between 10 and 100 (print a total as well)
 - Prints all positive powers of two, 1, 2, 4, 8, ..., up to 2³⁰
 - Prints all even integers 2 thru 200 on 10 different lines (10 numbers per line) in reverse order
 - Prints the following pattern of numbers (hint: use some value of i+10j):

```
11, 21, 31, 41, ..., 91, 101
12, 22, 32, 42, ..., 92, 102
...
20, 30, 40, 50, ..., 100, 110
```

- Write a program to project the total earnings in a savings account with a fixed APR, initial balance, and monthly deposit over a specified number of years.
- Implement a program to compute an estimate of the integral of some function using Simpson's rule of adding up small intervals of the value of the function
- Implement a program to estimate e (the exponential) by using the rule that $e = \sum_{k=0}^{\infty} 1 / k!$
- Implement a program to use the Babylonian method to compute a square root of a number a using the series,
$$x_{n+1} = .5 * (x_n + a / x_n), x_0 = 1$$