
CISS240: Intro to Programming Lecture: The End

Yihsiang Liow

What Have We Covered?

- Input/output: cin, cout
- Comments:
 - `/*...*/`
 - `//`
- Identifiers: naming convention, alphanumerics, cannot begin with numeric, do not begin with underscore, etc.
- Variables
- Basic data types: int, char, bool
- Arrays:
 - 1-dim, 2-dim, arrays and pointers
 - Array of characters as string terminated by '\0'

What Have We Covered?

- ♦ structs
- ♦ enum
- ♦ typedef: give new name to a type
- ♦ Statements, block of statements
- ♦ Operators:
 - ♦ `+, -, *, /, %, ==, !=, >=, >, <, <=, &&, ||, !`
 - ♦ Precedence

What Have We Covered?

- ♦ Selection:
 - ♦ if
 - ♦ If-else
 - ♦ Ternary ?:
 - ♦ switch-case
- ♦ Iterations:
 - ♦ for
 - ♦ while
 - ♦ do-while
- ♦ Goto: avoid using goto

What Have We Covered?

- ♦ Memory:
 - ♦ linear arrays of cells
 - ♦ two things associated with each cell: address and value
- ♦ Pointers:
 - ♦ new and delete
 - ♦ Drawing the memory model
 - ♦ De-referencing a pointer using *
 - ♦ Assigning address using &
 - ♦ Memory leak
 - ♦ Free store (or memory heap)

What Have We Covered?

- ♦ References:
 - ♦ Alias, must be initialized
 - ♦ Use for pass-by-reference to functions if function wants to modify variable passed in, or to speed up things
 - ♦ If function should not modify variable passed in by ref must use const
- ♦ Stack:
 - ♦ Think of this as a post office between functions
 - ♦ Think of this as a place for passing values to functions and for keeping return values

What Have We Covered?

- ♦ Functions:
 - ♦ Header, prototype
 - ♦ pass by value (including pointers)
 - ♦ pass by reference (including const ref)
 - ♦ int/char/bool/struct are passed by value
 - ♦ Array also pass by value but arrays are pointers so actually pointers are passed in, so function can modify elements of array
 - ♦ Inline
 - ♦ Template
 - ♦ Recursive functions

What Have We Covered?

- ♦ Scope:
 - ♦ Variables declared in a $\{...\}$ is not available outside that $\{...\}$
 - ♦ If x is declared before a $\{...\}$ and also in the $\{...\}$, then in $\{...\}$ x refers to the x in the block
- ♦ Some important algorithms/techniques:
 - ♦ Monte-Carlo simulation
 - ♦ Bubble sort
 - ♦ Binary search
 - ♦ Eratosthenes sieve

What Have We Covered?

- ♦ Some useful functions:
 - ♦ Math: srand, time, rand, pow, etc.
 - ♦ Format: setprecision, etc.
 - ♦ String: strcpy, strlen, etc.
- ♦ Different types of errors:
 - ♦ syntax error: not using language correctly, error occurs while compiling program
 - ♦ logic error: program works but give wrong results!
 - ♦ runtime error: no error while compiling but error occurs while running program
- ♦ Program tracing

What now ... ?

- ♦ What now ... ?
- ♦ From here you can go onto CISS 245 where we will study object-oriented programming (OOP)
- ♦ OOP is very new and very important (good for the resume!)

What now ... ?

- You can easily pick up other (imperative) programming languages (most of them look similar!)

- C/C++, Java:


```
int s=0;
for (int i=1; i<5; i++)
    s = s + i;
```

- Python:


```
s = 0
for i in [1,2,3,4,5]:
    s = s + i
```

- Pascal:


```
var i: integer
var s: integer
s := 0
for i := 1 to 5 do
    s := s + i
```

What now ... ?

- ♦ You also need to know more problem solving techniques:
 - ♦ Mathematics
 - ♦ Discrete Math, Algorithms, Automata theory, Cryptography, ...
 - ♦ Ability to design software
 - ♦ OOAD
 - ♦ Software engineering
 - ♦ Operating system, networks
 - ♦ Hardware
 - ♦ Etc.

What now ... ?

The Road goes ever on and on
Down from the door where it began.
Now far ahead the Road has gone,
And I must follow, if I can,
Pursuing it with eager feet,
Until it joins some larger way
Where many paths and errands meet.
And whither then? I cannot say.

-- J R R Tolkien
