* + 1. How to Learn C++  
       -  
       Getting yourself   
       and your workspace ready

# Programming Languages

* Are human languages
  + Have roots in older languages
  + Change over time
  + Adopt useful features from competing languages
  + Have recognizable periods of use

|  |  |  |
| --- | --- | --- |
| **Era** | **English** | **C++** |
| Root Language | Celtic | C |
| Ancient | Old English | C with Classes |
| Classical | Middle English | C++98, C++03 |
| Early modern | Elizabethan | C++11 |
| Modern | Modern | C++14 |
| Modern | Hipster | C++17 |
| Post-Modern | Newspeak | C++20 |

* Many local dialects and conventions
  + Usage often constrainted by context (e.g. MISRA rules, "Polite" english)
  + We take language very personally
  + We love to bicker about details
* Some ancient forms survive
  + Verily, Thou knowest this to be naught but Truth
  + x = \*(c\_ptr + 1);
* Humans are EXTREMELY GOOD at learning languages
  + You are hard-wired for this!

# You Won't Just Learn C++

* You also get these Bonus Features!
* Command shell(s)
  + A chance to meet a hundred new commands
* Editor(s)
  + Make it your own with fun extensions and customizations galore!
* Toolchain (preprocessor, compiler, linker)
  + A million options for a million needs!
* Git
  + Killer or Quirky? You decide! It's here to stay.
* Build tools
  + Wrestle your toolchain to see who's boss
* Testing tools
  + Whose rules really rule?
* Run-time platforms
  + You want architecture? We got it. Also beaucoup hardware.

# Compartmentalize

* You will switch contexts a lot
* Cheatsheets and checklists are essential
  + I'll give you some; expand them and make your own
* Build your environment to help you maintain contexts
  + Reserve a place to code and study
  + Include visual and sensory cues that mark your "study" space
  + Use visual cues to distinguish your "coding" apps
* Organize your virtual desktop
  + Put course materials in a folder that is easy to get to
  + Organize labs and handouts by subject, module #
  + Name folders and files to help your future self find them
  + Tidy up screen backgrounds and toolbars
* Organize your physical space
  + Room for books, manuals, notebooks
  + Room to stretch
  + Room to write

# Learning C++ Syntax

* There's a lot of muscle memory involved
  + Learned by repetition, practice
* Slow down
  + Read/write word by word
* Use it, don't avoid it
  + Write some things over and over until they feel right
  + Spell out words completely many times before relying on intellisense
  + Read the references even when you don't completely understand them
* Consolidate your Memories
  + Read and repeat concepts out loud
  + Say it in other words
  + Writing it down
  + Tell someone else about it
  + Take a break
  + Sleep on it
* Consult expert viewpoints to try to understand WHY
  + It'll make you laugh, after it makes you cry

# Keep Us Honest

* Verify what we teach by trying it!
  + Test it, try to break it
* You'll be Googling a LOT!
  + And I will point you to lots of resources
* Check more than one source
  + There's usually several accepted ways to accomplish the same thing
    - Old C way
    - Classic C++ way
    - Modern C++ way
  + Plus, several idiosyncratic ways that are one person's dream, another's nightmare
* Ask for others' opinions; don't be surprised by strong feelings
  + Better have a sense of humor
  + Be adaptable
  + Be resilient

## Don't rely on "Tips and Tricks"; internalize the language!

# Work with a Net

* Examples
* References
* Books
* Fellow programmers
* Your instructor
* Code Management repositories and Backups
* IntelliSense / Code completion
* Debugger

# Get Comfy

* We'll be doing a LOT of typing
  + But you'll be doing 10x more sitting, reading, and talking
* Eat, hydrate, take bathroom breaks
* Stand up and stretch!
* Have snacks and bevs handy
* Get enough sleep

# Ergonomics

* Decent lighting
* A comfortable chair
* Keyboard and monitor at the correct heights
* A good keyboard and mouse, easy to hold, tactile feedback
* Large screen(s) and big, clear fonts
* A pleasant, context-establishing windows theme
* Physical aids as appropriate
  + Reading glasses
  + Footrest, armrest

# Be the Protagonist

* Don't be a Hero
  + Works alone
  + Remote and Admirable
  + Needs superpowers
  + Has enemies
  + Can't afford to make mistakes
  + Lives and dies by their power
  + Often dies tragically, romantically, heroically!
* Protagonist
  + Collaborates
  + Close and Understandable
  + Has human powers
  + We're all rooting for you
  + Learns by doing, sometimes has "happy accidents"
  + Gets caught up in exciting things and has adventures
  + Often has happy endings

# Don't Worry

* There's plenty for everyone
* Ask for help
* You can't break the machines
  + You can't break my backup repository
* You can't fail
* I already made all the mistakes I could think of
  + I left some for you so you woudn't feel left out
* Savor your successes

# Things you think you'll NEVER get, but you will!

* Pointers
* Memory management
* Thread synchronization
* Templates

# Analyzing a Problem

* Take time to Think
  + Go slow
  + Don't start coding immediately
* Make a plan
  + Decide WHAT you want as an end result
  + Analyze HOW to get it
  + Consider WHY you should choose one way over another
* Pick problem apart and think about ONE small, concrete thing
  + Big puzzles are made up of little pieces - find one
  + Writing one tiny function can help crystallize ideas
* Use your crafty skills
  + Use //comments to brainstorm and organize program flow
  + Draw a sketch or a diagram
  + Make a physical model
  + Make it personal - find an analog in human interactions or your own life
* Be on the lookout for similar problems
  + Have I seen something like this before?
  + Look at other people's code
* Don't worry about "elegant" and "efficient" until you have to
* Throw Stuff Away