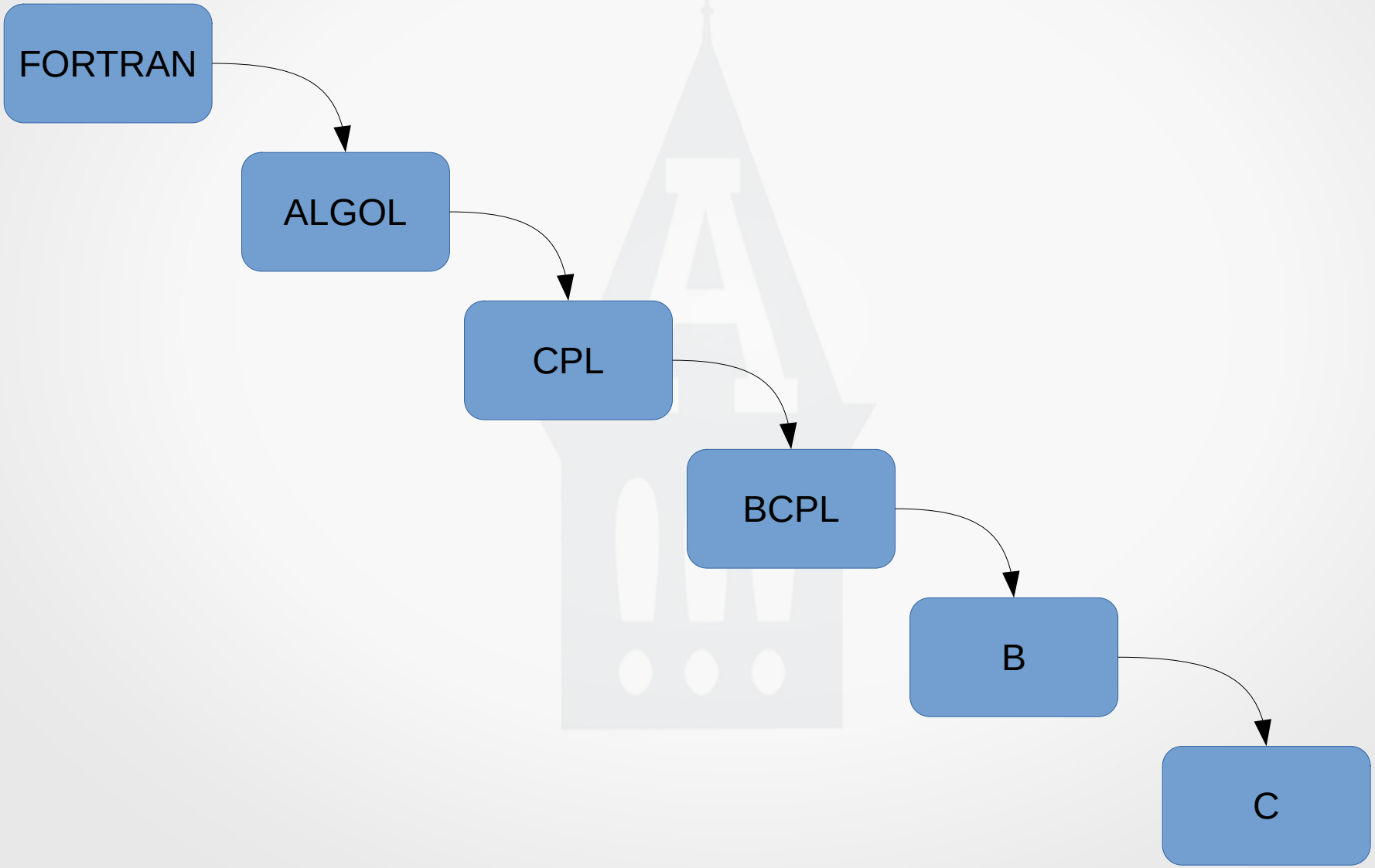
CS 3460

History of C++

Lineage of C++



Lineage of C++

- FORTRAN
  - 1957
  - John Backus
- ALGOL 58/60
  - 1958/1960
  - Many designers
- CPL (Common/Cambridge Programming Language)
  - 1963
- The BCPL language (Basic CPL)
  - Intended for writing compilers.
    - The BCPL compiler was written in BCPL.
  - Martin Richards, University of Cambridge
  - First brace { } programming language, along with single line comments using // (dropped by C, added back in C++)
  - First Hello World implementation
- The B language
  - Bell Labs around 1969
  - Ken Thompson and Dennis Ritchie
- The C language
  - Bell Labs, 1972
  - Dennis Ritchie

Lineage of C++

- FORTRAN
  - 1957
  - John Backus
- ALGOL 58/60
  - 1958/1960
  - Many designers
- CPL (Common/Cambridge Programming Language)
  - 1963
- The BCPL language (Basic CPL)
  - Intended for writing compilers.
    - The BCPL compiler was written in BCPL.
  - Martin Richards, University of Cambridge
  - First brace { } programming language, along with single line comments using // (dropped by C, added back in C++)
  - First Hello World implementation
- The B language
  - Bell Labs around 1969
  - Ken Thompson and Dennis Ritchie
- The C language
  - Bell Labs, 1972
  - Dennis Ritchie

```
procedure  Transpose (a) Order: (n);
value  n;
array  a;
integer  n;
begin  real w; integer  i, k;
for  i := 1 step 1 until  n do
    for  k := 1+i step  1 until n do
        begin
                w:=a[i,k];
                a[i,k]:=a[k,i];
                a[k,i]:=w
        end
end Transpose
```

Lineage of C++

- FORTRAN
  - 1957
  - John Backus
- ALGOL 58/60
  - 1958/1960
  - Many designers
- CPL (Common/Cambridge Programming Language)
  - 1963
- The BCPL language (Basic CPL)
  - Intended for writing compilers.
    - The BCPL compiler was written in BCPL.
  - Martin Richards, University of Cambridge
  - First brace { } programming language, along with single line comments using // (dropped by C, added back in C++)
  - First Hello World implementation
- The B language
  - Bell Labs around 1969
  - Ken Thompson and Dennis Ritchie
- The C language
  - Bell Labs, 1972
  - Dennis Ritchie

```
procedure  Transpose (a) Order: (n);
value  n;
array  a;
integer  n;
begin  real w; integer  i, k;
for  i := 1 step 1 until  n do
    for  k := 1+i step  1 until n do
        begin
                w:=a[i,k];
                a[i,k]:=a[k,i];
                a[k,i]:=w
        end
end Transpose
```

```
v[2000];
n 2000;

main() {
    extrn v, n;
    auto i, c, col, a;

    i = col = 0;
    while (i < n)
        v[i++] = i;
}
```

C++

- C with Classes
  - 1979
  - Bjarne Stroustrup, PhD work
  - Class concept inspired from Simula
- C++
  - 1983
  - Inheritance, polymorphism, stronger type system
- 1985 – First Commercial Release of C++
  - CFront
  - Translated C++ to C, then used a C compiler to create an executable
- 1989 – Version 2.0 of the language
  - Multiple inheritance, protected access, abstract classes, new/delete operators
- 1998 – First ISO C++ standard released
  - Known as C++98
  - Boolean type, exceptions, templates, namespaces
  - C++ Standard Template Library

“Modern” C++

- Until 2011, the language, languished (alliteration intended)
- 2011 – C++11 standard ratified
  - Took several years for compiler vendors to catch up
  - Huge step forward, new beginning for the language
  - inferred data types, constant expressions
  - lambdas, move operations, range-based loops
  - initializer lists, concurrency, smart pointers, regex
  - ...and more…

“Modern” C++

- 2014 – C++14; minor update
  - variadic templates, return type deduction
  - digit separators, generic lambdas
- 2017 – C++17; modest update
  - initializers in `if` and `switch` statements
  - improved auto type deduction, compile-time static if
  - nested namespace definitions, structured bindings
- 2020 – C++20; huge update
  - modules, concepts, ranges, `std::format`, and more

“Modern” C++

- 2023 – C++23; modest update
  - Removed some legacy garbage collection support
  - Multidimensional subscript operator
  - Literal suffixes for size_t and ptrdiff_t
  - contains() member for string/string_view
  - std::expected
  - <stacktrace> library for debugging support
  - std::print to send formatted output to stdout
  - std::generator for better coroutine support
  - A whole bunch of other “quality of life” improvements
  - Compiler support isn’t fully there yet

“Modern” C++

- 2026 – C++ 26
  - Reflection
  - Contracts
  - Improved string and string_view
  - Ranges improvements
  - Linear algebra support
  - Additional debugging support
  - Much more already defined
  - More still to come, very fluid design at this point

C++ Relationship to C

- C++ is not a strict superset of C
  - C allows variable length *raw* arrays, C++ does not
  - C allows implicit conversion of `void*` to other types, C++ does not
  - C has a `restrict` keyword that C++ does not
  - C++ reserves additional keywords like `new` and `delete`, while these can be used as identifiers in C