Intro to C

CS 224

Chris Archibald

C: Background

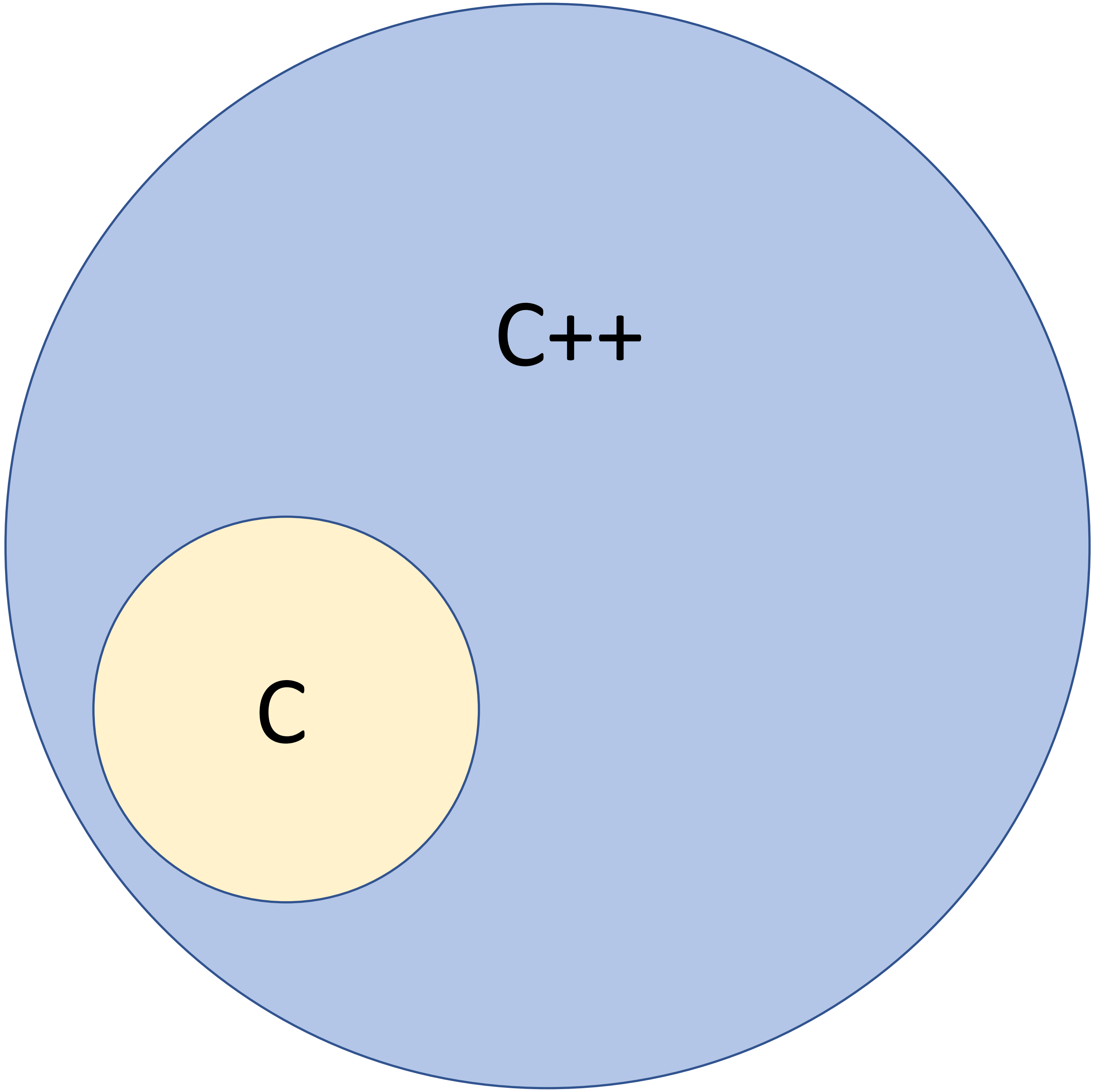
- Came after the language B
- C is “quirky, flawed, and an enormous success”
- C closely tied to Unix operating system
- C is small, simple language
  - Design controlled by single person
- C designed to implement the Unix operating system
  - “Low level” language - language of choice for system-level programming
  - Replaced using assembly for these tasks

Quotation

“Since C is relatively small, it can be described in a small space, and learned quickly. A programmer can reasonably expect to know and understand and indeed regularly use the entire language.”

The C Programming Language, Kernighan & Richie

C: in relation



C: Data Types

- Whole number types (differ in amount of space/memory)
  - `char` : smallest – enough space to:
    - represent small numbers (256 different values)
    - represents single characters ‘A’, ‘e’, ‘;’, etc.
  - `short`: (next biggest) short integer (fewer possible values)
  - `int` : (normal size) integer numbers (whole numbers)
  - `long` : long integer (more possible values) - (bigger size = more numbers)
- Floating point numbers –
  - These are numbers with fractional parts (i.e. 3.14159)
  - `float` : normal size
  - `double` : high precision floating point (more decimal places)

C: Control flow

- Same as C++
- if-else:

```
if (expression) {
        statement
}
else {
        statement
}
```

C: Control flow – loops:

```
while ( expression ) {
   statement
}

for (expr1 ; expr2; expr3 ) {
   statement
}
```

C: Control flow - switch

```
switch ( expression ) {
   case const-expr:
     statements
   case const-expr:
     statements
   default:
     statements
}
```

STDOUT

- `cat` and `echo` write to STDOUT from command line
- In C++: `cout << “Hello”;`
- In C: `printf(“Hello”);`
- Outputting a variable (say an integer num)
  - In C++: `cout << “This is num: “ << num;`
  - In C: `printf(“This is num: %d\n”, num);`
  - Format strings
    - `%c` – character
    - `%d` – decimal integer
    - `%x` – hexadecimal integer
    - `%f` – floating point number
    - `%l` – double
    - `%s` – character string (array of `chars`)
    - Many other options as well (significant digits, decimal places, etc)

STDIN

- In C++ you used: `cin >> num;`
- In C we use: `scanf( “%d”, &num);`
- Other format strings:
  - `%c` – character
  - `%d` – decimal integer
  - `%x` – hexadecimal integer
  - `%f` – floating point number
  - `%l` – double
  - `%s` – character string (array of `chars`)

Let’s write some code!