

CSC 211: Computer Programming

Introduction to C/C++

Michael Conti

Department of Computer Science and Statistics
University of Rhode Island

Summer 2024



Original design and development by Dr. Marco Alvarez

Administrative notes

Administrative notes

- A00 Groups have been assigned
 - Due 02/04/2024
- Lab#01
 - Begin next week
- Are you on Piazza?
 - Get the app!
- Are you on Gradescope?
- Discussion sessions Next Week
- Communication Preference
 - Piazza Only

Algorithms and Programs

Problems, Algorithms and Programs

• Problem

- ✓ task to be performed (precisely defined)
- ✓ well-defined **inputs** and **outputs**
- ✓ may include constraints

• Algorithm

- ✓ set of concrete steps required to solve a problem
- ✓ properties:
 - it must be correct (must compute the desired function)
 - it is composed of a series of concrete and finite number steps
 - there can be no ambiguity as to which step will be performed next
 - it must terminate

5

Problems, Algorithms and Programs

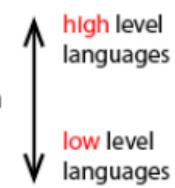
• Program

- ✓ instantiation of an algorithm using a programming language

Snap, Scheme, Prolog, Lisp

JavaScript, Python, Java, Alice, Scratch

C, C++



<https://bjc.edc.org/bjc-r/cur/programming/6-computers/1-abstraction/03-software-languages.html>

6

Example

An Algorithm

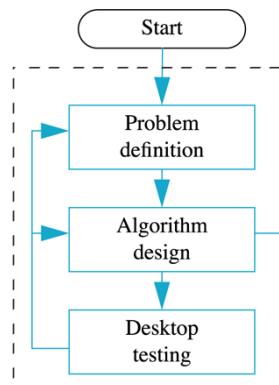
**Algorithm that determines how many times
a name occurs in a list of names:**

from: Problem Solving with C++, 10th Edition, Walter Savitch

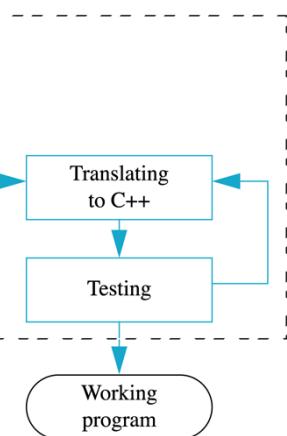
7

Program Design Process

Problem-solving phase



Implementation phase



from: Problem Solving with C++, 10th Edition, Walter Savitch

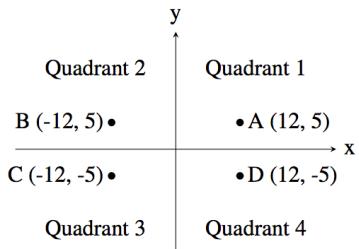
8

Example

Read a point from user and determine the quadrant it is in. You can assume that neither of the two coordinates will be 0

```
read first number into x
read second number into y
if x and y are positives
    print "Quadrant 1"
else if x is positive and y is negative
    print "Quadrant 4"
else if x is negative and y is negative
    print "Quadrant 3"
else
    print "Quadrant 4"
```

<https://open.kattis.com/problems/quadrant>



9

Example (program)

```
# read numbers
X = input('Enter first number: ')
y = input('Enter second number: ')

# perform selection
if x > 0 and y > 0:
    print('Quadrant 1')
else if x > 0 and y < 0:
    print('Quadrant 4')
else if x < 0 and y < 0:
    print('Quadrant 3')
else:
    print('Quadrant 2')
```

10

Example (program)

```
#include <iostream>

int main() {
    // read numbers
    int x, y;
    std::cout << "Enter first number: ";
    std::cin >> x;
    std::cout << "Enter second number: ";
    std::cin >> y;
    // perform selection
    if (x > 0 && x > 0) {
        std::cout << "Quadrant 1\n";
    }
    else if (x > 0 && y < 0) {
        std::cout << "Quadrant 4\n";
    }
    else if (x < 0 && y < 0) {
        std::cout << "Quadrant 3\n";
    }
    else {
        std::cout << "Quadrant 2\n";
    }
}
```

<https://godbolt.org/z/oFwd6N>

11

C/C++

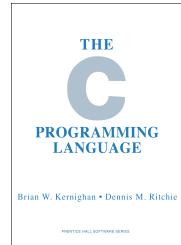
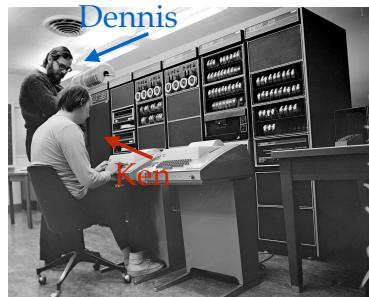
History

- Ken Thompson created the B language while developing UNIX (implemented in assembly) at Bell Labs [1970]

- ✓ slow and interpreted

- Dennis Ritchie began development of a compiler for B and could produce executable code [1972]

- ✓ became known as the C language
 - ✓ Linux kernel reimplemented in C

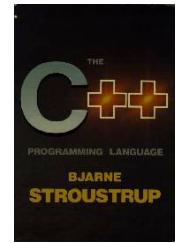
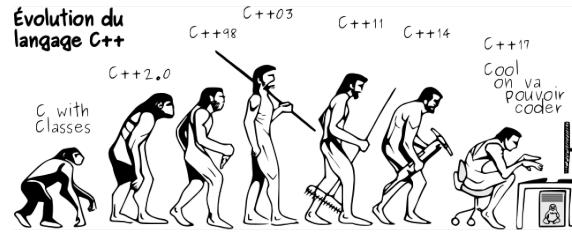
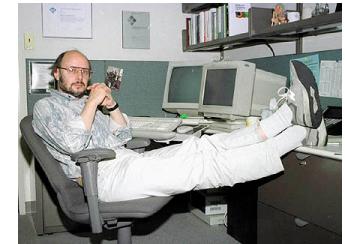


13

History

- Bjarne Stroustrup began the development of C++ (also from Bell Labs) [1980]

- ✓ object oriented, generic, functional



14

C++?

- Static type system
 - ✓ prevents unintended operations
 - ✓ optimized machine code (i.e. faster and/or using less memory)

- Object oriented language
 - ✓ improves maintainability

- When to use it?
 - ✓ performance matters
 - ✓ developing time is less important
 - ✓ specialized libraries require it

C/C++?

- Pros

- ✓ vast documentation freely available
 - ✓ provides different levels of abstraction (from data structures to memory management)
 - ✓ it is compiled
 - ✓ high performance

- Cons

- ✓ steep learning curve
 - ✓ large language
 - ✓ no automatic memory management (can be an advantage)
 - ✓ requires attention to minor details
 - ✓ GUIs only available through extensive libraries (less portable)

15

16

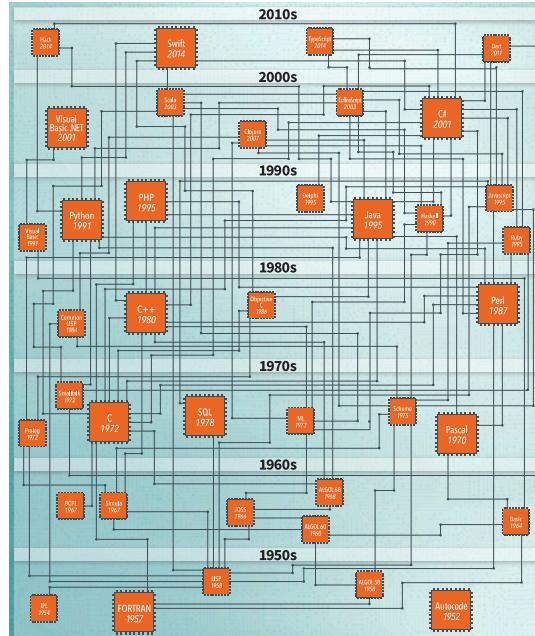
Console applications

The screenshot shows the CS50 IDE interface. At the top, there's a menu bar with File, Edit, Find, View, Go. Below it is a toolbar with icons for workspace, settings, share, collaborate, outline, and debugger. The main area displays the code for 'hello.c':

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("hello, world\n");
6 }
```

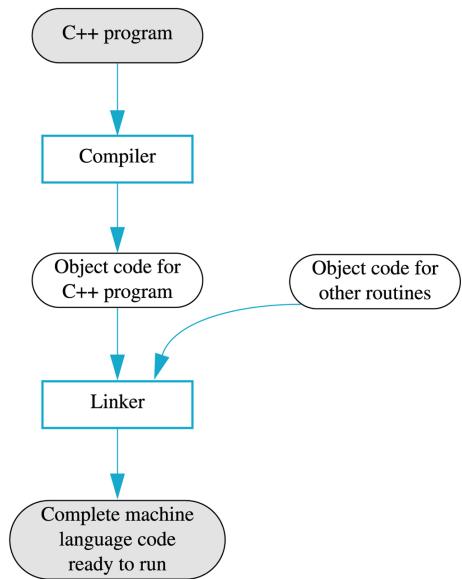
At the bottom, there's a terminal window showing the command `~/workspace/ $`.

17



18

Preparing a C++ Program for Running

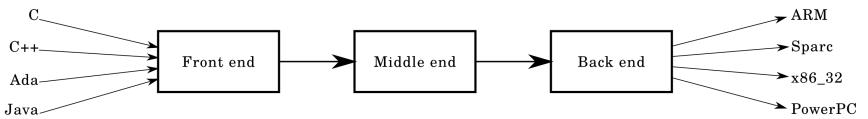


from: Problem Solving with C++, 10th Edition, Walter Savitch

19

Compilers

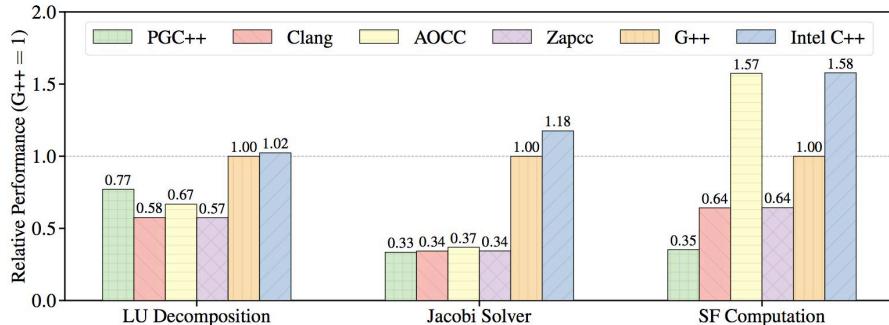
- A computer program that ...
 - ✓ translates source code from one programming language to another (usually from high-level to low-level languages)
 - ✓ performs code optimizations
 - ✓ provides error checking



Correctness is paramount. Compilers cannot afford to fail.

20

C++ Compilers



single-threaded, higher is better

21

```
#include <iostream>
```

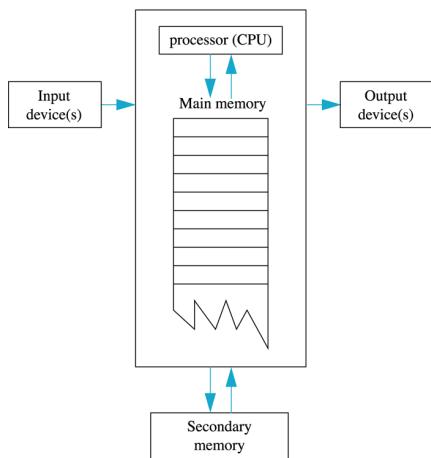
```
int main()
{
    std::cout << "Hello World!" << std::endl;
    return 0;
}
```

```
~/workspace/ $ g++ hello.cpp -o hello
~/workspace/ $ ls -l
total 16
-rwx----- 1 ubuntu ubuntu 9176 Sep 10 15:21 hello*
-rw------- 1 ubuntu ubuntu     91 Sep 10 15:20 hello.cpp
~/workspace/ $ ./hello
Hello World!
~/workspace/ $
```

22

How programs run?

Main Components of a Computer



from: Problem Solving with C++, 10th Edition, Walter Savitch

23

CSC 211: Computer Programming Introduction to C/C++

Michael Conti

Department of Computer Science and Statistics
University of Rhode Island

Summer 2024



Original design and development by Dr. Marco Alvarez