CSC 211: Object Oriented Programming

Introduction to C/C++

Michael Conti

Department of Computer Science and Statistics
University of Rhode Island

Fall 2019

Original design and development by Dr. Marco Alvarez

Administrative notes

TA Help

Labs

Monday
John, Tanner, Jordan (3:45p - 4:45p)

Thursdays
Rodrigo, Jordan, Christian (3:30-4:30)

Office Hours

Monday

Tuesday
Christian: 11:00a – 2:00p, 3:15p – 5:00p
Rodrigo: 12:30 - 2:00pm, 3:15p - 5:00p

Wednesday
Christian: 12:00p – 4:15p

Thursday
Christian: 11:00a – 2:00p
Rodrigo: 12:30p - 1:30p

Friday

Discussion Section

Tuesday 5:00 – 6:00 (Christian)
Wednesday 4:20 – 5:20 (Christian)

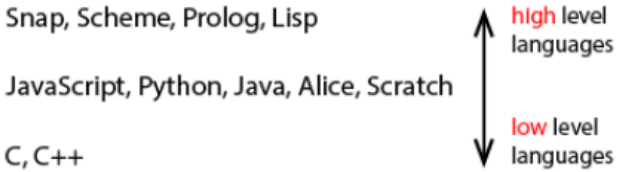
Algorithms and Programs

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

Problems, Algorithms and Programs

- Program
  - instantiation of an algorithm using a programming language



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

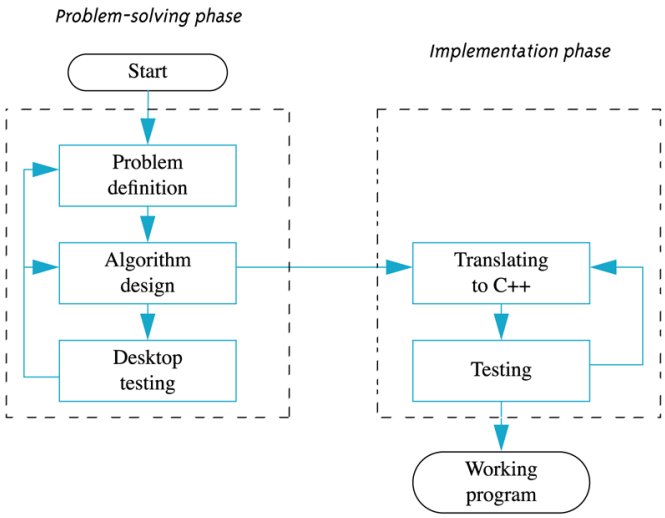
Example

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

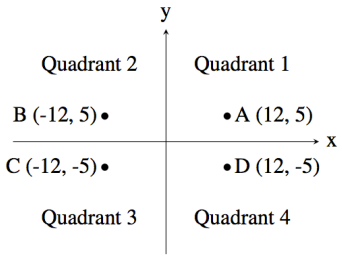
Program Design Process



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

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 num1
read second number into num2
if num1 and num2 are positives
    print "Quadrant 1"
else if num1 is positive and num2 is negative
    print "Quadrant 2"
else if num1 is negative and num2 is negative
    print "Quadrant 3"
else
    print "Quadrant 4"
```

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

Example (program)

```python
# read numbers
num1 = input('Enter first number: ')
num1 = input('Enter second number: ')

# perform selection
if num1 > 0 and num2 > 0:
    print('Quadrant 1')
else if num1 > 0 and num2 < 0:
    print('Quadrant 4')
else if num1 < 0 and num2 < 0:
    print('Quadrant 2')
else
    print('Quadrant 3')
```


Example (program)

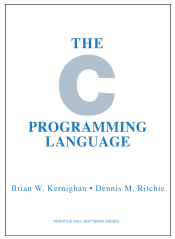
```cpp
#include <iostream>

int main() {
    // read numbers
    int num1, num2;
    std::cout << "Enter first number: ";
    std::cin >> num1;
    std::cout << "Enter first number: ";
    std::cin >> num2;
    // perform selection
    if (num1 > 0 && num2 > 0) {
        std::cout <<  "Quadrant 1\n";
    }
    else if (num1 > 0 && num2 < 0) {
        std::cout <<  "Quadrant 2\n";
    }
    else if (num1 < 0 && num2 < 0) {
        std::cout <<  "Quadrant 3\n";
    }
    else {
        std::cout <<  "Quadrant 4\n";
    }
}
```

https://godbolt.org/z/OFwd6N

C/C++

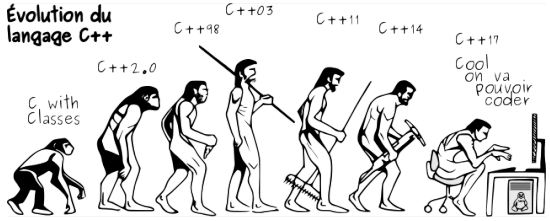
History

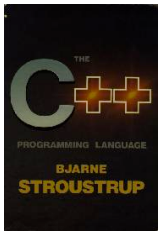
- Ken Thompson created the B language while developing UNIX (implemented in assembly) at Bell Labs [1970]
  - slow and interpreted
- Dennis Ritchie began development of a compiler for B and could produce executable code [1972]
  - became known as the C language
  - Linux kernel reimplemented in C



History

- Bjarne Stroustrup began the development of C++ (also from Bell Labs) [1980]
  - object oriented, generic, functional





https://github.com/cpp-frug/materials/tree/gh-pages/images

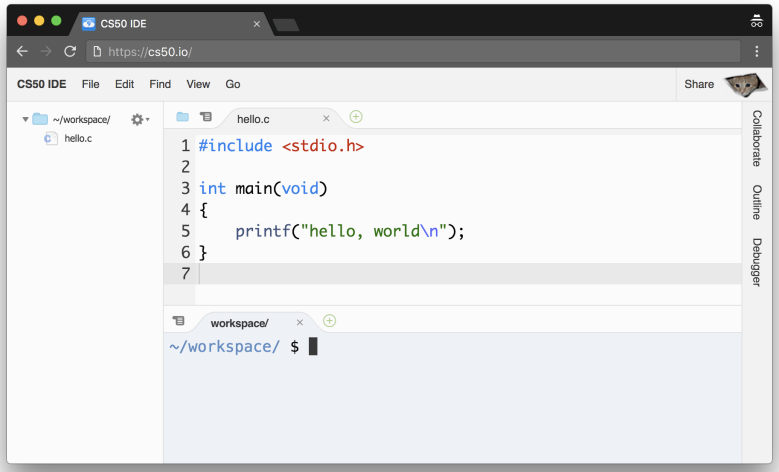
C++?

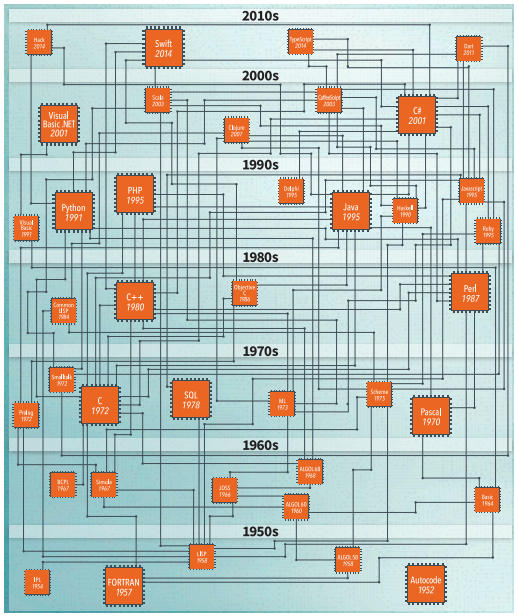
- Static type system
  - prevents unintended operations
  - optimized machine code (i.e. faster and/or using less memory)
- Object oriented language
  - improves maintainability
- When to use it?
  - performance matters
  - developing time is less important
  - specialized libraries require it

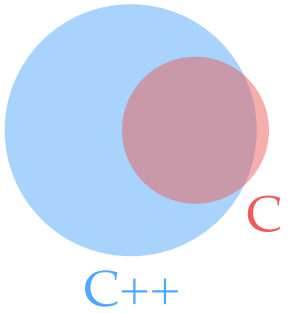
C/C++?

- 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)

Console applications

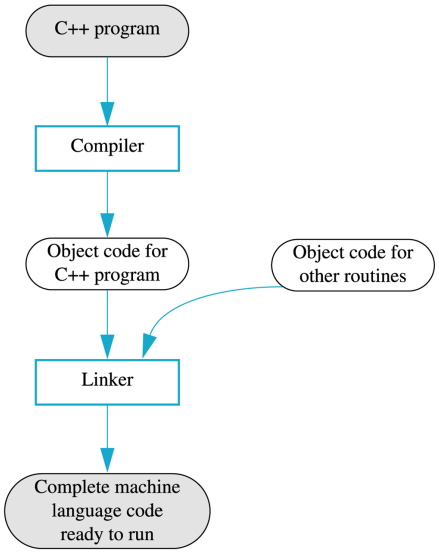






https://www.thesoftwareguild.com/blog/history-of-programming-languages/

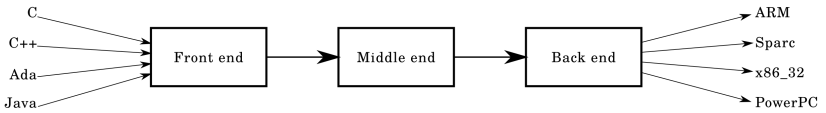
Preparing a C++ Program for Running



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

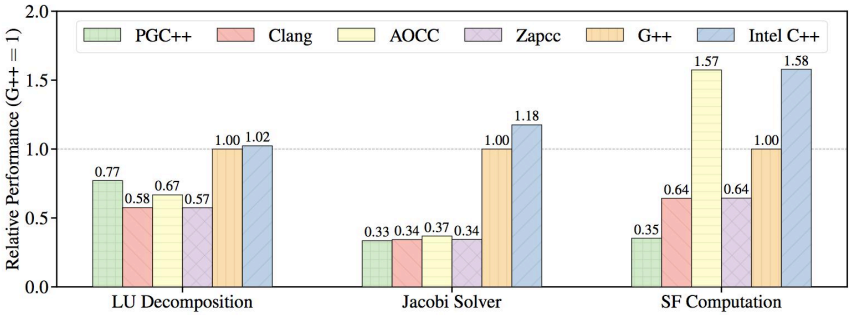
Compilers

- 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.`

C++ Compilers

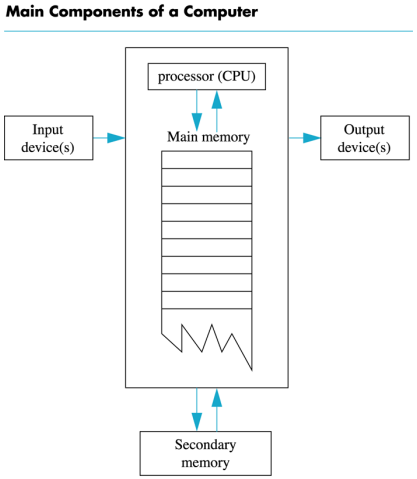


single-threaded, higher is better


```cpp
#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/ $
```


How programs run?



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