

Math 5370: Transitioning to C++ for Scientific Computations

Dr. Natasha Sharma

25th and 27th August 2015

Agenda for the week

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

1 Computer Systems

2 Programming and Problem-Solving

3 Introduction to C++

- Hardware
- Software
- High-Level Languages
- Compilers

Hardware

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

Whether working on a personal computer, a workstation or a mainframe, the hardware is made up of 5 main parts:

- Input device (keyboard, mouse)
- Output device (monitor, printer)
- Processor (Central Processing Unit)
- Main memory
- Secondary Memory

More on Memory: Main/ Primary Memory

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

The C++ program that is executed is stored in the main memory in the form of long lists of numbered locations.

Each location contains a combination of 0s and 1s.

A digit which take these two particular values are called **binary digit** or **bit**.

A byte is a unit of data that is eight binary digits long. A byte is the unit most computers use to represent a character such as a letter, number, or symbol (for example, "g", "5", or "?").

Main memory also referred to as random access memory since the computer can access the data in any location.

More on Memory: Secondary Memory

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

The C++ program is essentially a list of instructions which is living in the main memory.

The Secondary memory is where programs and data are kept on a more permanent basis example hard disk. However, unlike the main memory, accessing this involves copying it to the main memory first.

Software

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

While the hardware is the physical component of the computer, the software is essentially the set of instructions which is understandable by the computer and controlling the use of the processor.

One of the most important software is the operating system which manages the hardware and software as well as provides a means to communicate to the computer.

High Level and Low Level Language

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

The set of instructions written in the human language defines a High Level Language.

While a Low Level language is one which is comprehensible by the computer.

Example: **ADD C D**

This low level instruction can be interpreted by the computer by first changing the **ADD** to a combination of bits say 1000
C and **D** could be 0110 and 1001 respectively.

High level to machine language

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

The program which translates a high level language to a machine language is called a compiler.

Example: A compiler for translating the C++ program to the machine language is **Cygwin (GNU C++)** for Windows.

We provide a **source code** (say `hello_world.cpp`) as an input to the compiler `g++` and what it spits out is the **object code** (given a default name `a.out`).

Command for compiling the code:

```
g++ hello_world.cpp
```

The object code must be combined with the object code of other routines such as input output routine. This is called **linking**.

Programming and Problem-Solving : Algorithm

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

Think about the task of finding the roots of the quadratic equation:

$$ax^2 + bx + c = 0.$$

If we are lazy about it and want to use a calculator for all the operations involved, based on the most basic calculator used, we can reduce this task to merely performing the operations of multiplication, square root, addition and division.

Now if we are more lazy about this task and want to recursively use the formula for root finding with only a change of a , b and c , we would be required to provide a set of instructions.

Algorithm

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

- Provide the three numbers a, b, c .

- Compute $d = \sqrt{b^2 - 4ac}$

- The two roots are r_1 and r_2 where

$$r_1 = \frac{-b + d}{2a}, \quad r_2 = \frac{-b - d}{2a}.$$

Algorithm

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

- Provide the three numbers a, b, c .

- Compute $d = \sqrt{b^2 - 4ac}$

- The two roots are r_1 and r_2 where

$$r_1 = \frac{-b + d}{2a}, \quad r_2 = \frac{-b - d}{2a}.$$

Algorithm

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

- Provide the three numbers a, b, c .

- Compute $d = \sqrt{b^2 - 4ac}$

- The two roots are r_1 and r_2 where

$$r_1 = \frac{-b + d}{2a}, \quad r_2 = \frac{-b - d}{2a}.$$

Algorithm

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

- Provide the three numbers a, b, c .
- Compute $d = \sqrt{b^2 - 4ac}$
- The two roots are $r1$ and $r2$ where

$$r1 = \frac{-b + d}{2a}, \quad r2 = \frac{-b - d}{2a}.$$



$$a = 0 \quad ?$$

- In such a case the two roots will be a quantity identified as infinity or if the numerator $-b \pm d$ is also zero then, it is NaN (not a number).
As programmers, it is our responsibility to eliminate such cases !
- Kind of like the algorithm on the next slide !



$$a = 0 \quad ?$$

- In such a case the two roots will be a quantity identified as infinity or if the numerator $-b \pm d$ is also zero then, it is NaN (not a number).
As programmers, it is our responsibility to eliminate such cases !
- Kind of like the algorithm on the next slide !

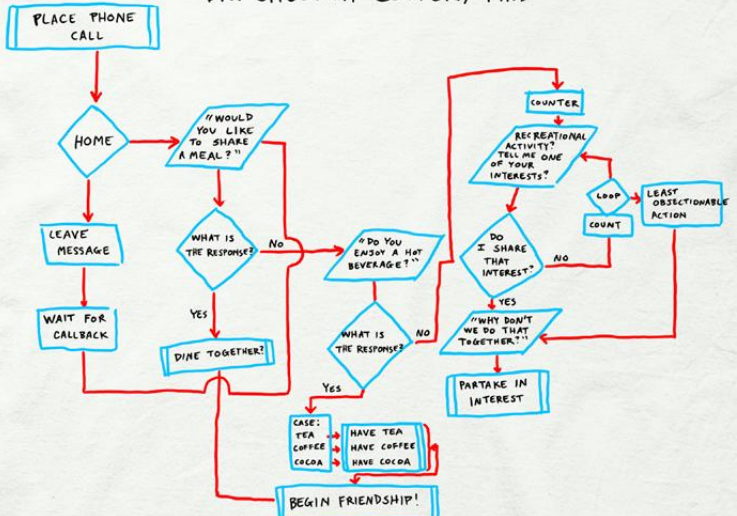


$$a = 0 \quad ?$$

- In such a case the two roots will be a quantity identified as infinity or if the numerator $-b \pm d$ is also zero then, it is NaN (not a number).
As programmers, it is our responsibility to eliminate such cases !
- Kind of like the algorithm on the next slide !

THE FRIENDSHIP ALGORITHM

DR. SHELDON COOPER, Ph.D



Origins of C++

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

- 1970s C was written by Dennis Ritchie (AT&T Bell Laboratories) for writing and maintaining the UNIX system.
- C gained popularity however, C had its own share of shortcomings for e.g. absence of runtime error checks and object oriented programming (oop).
- A decade later, Bjarne Stroustrup introduced C++ with one of the key features being the oop.
- Later versions of C++ are labeled C++11, C++14 etc.

my_first_code: Skeleton of the code

Math 5370:
Transitioning
to C++ for
Scientific
Computations

Dr. Natasha
Sharma

Computer
Systems

Programming
and Problem-
Solving

Introduction
to C++

```
#include <iostream>
using namespace std;

int main ()
{
    Variable Declarations

    Statement_1 ;
    Statement_2 ;
    ...
    Statement_n ;

    return 0 ;
}
```