CS 211 Notes

Introduction to Programming

Jaeden Bardati

September 22, 2021

1 Course Overview

September 10, 2021

1.1 Objectives

The course is intended to teach how to develop a computer program to solve a problem. C++ is a tools that will be used to develop these skills and logical thinking. These skills will be transferable to other languages.

2 Computer Organization

2.1 Hardware

September 11, 2021

2.1.1 Components

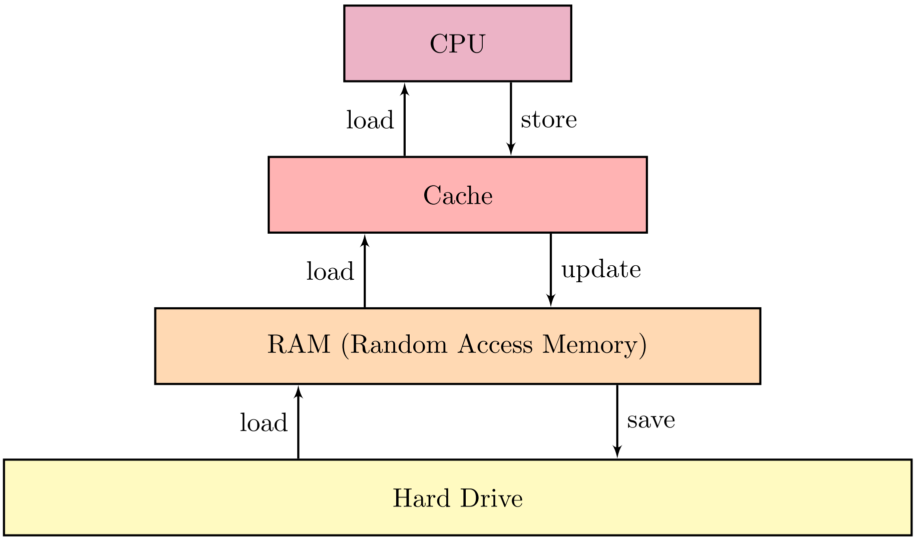
Modern computers are built using the **Von Neumann machine**. There are three aspects:

- **Architecture**: **I/O** (User interaction) + **Memory** (Storage) + **CPU** (CU: Control Unit, ALU: Arithmetic and Logic Unit). These are all connected by a shared bus.
- **Stored Programs**: All programs and data are stored in memory (binary).
- **Sequential Execution**: Also called the **fetch-decode-execute** cycle. Instructions are **fetched** from memory, **decoded** by the CU and then **executed** by the ALU. If there is a result, it is stored back in memory.

2.2 Memory

September 11, 2021

The memory is organized in a **hierarchy**. At the bottom of the hierarchy is the Hard Drive (in TB). At the top is the CPU. Since the hard drive is slow, when some data from the hard drive is needed, it is first loaded into **RAM (Random Access Memory)** (in GB). The RAM is still too slow for the RAM, so the data is stored in **cache** (in KB or MB). Yet still, this is not fast enough for the CPU, so **registers** (in Bytes) in the CPU itself are used to store variables.



As you **go up** the hierarchy, the **speed increases**, but the **size decreases** and the **cost increases**.

2.2.1 RAM

Random access memory is organized in an array of Bytes (”words”).

Words in RAM are addressed with a byte themselves (e.g. 01101101 is an address). These are typically written in hexadecimal (e.g. 6D).

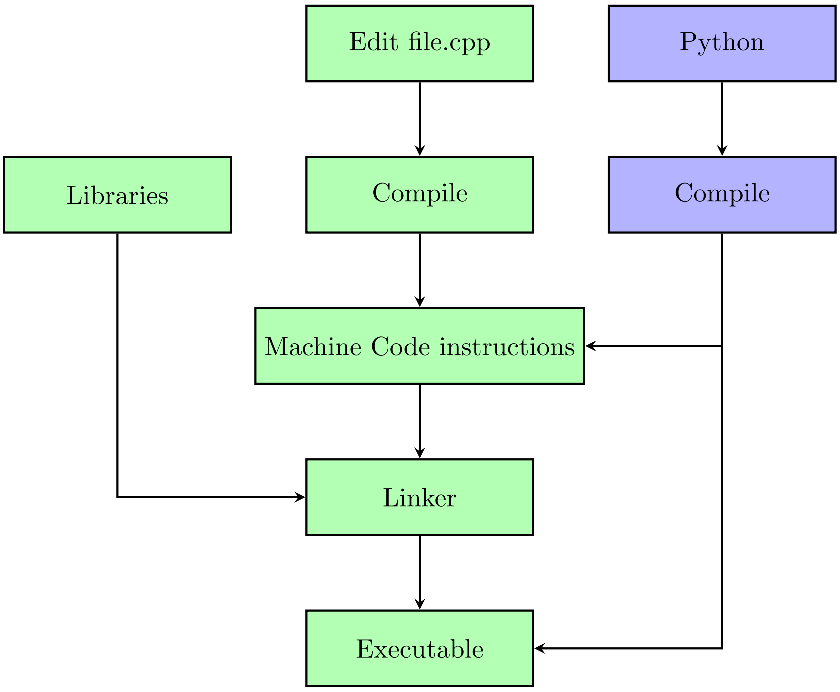
Words in RAM can be data or machine code instructions. Instructions contain a binary code for each operation (for example, addition). Instructions codes are dependent on the CPU.

3 Computer Organization

3.1 A Flow Chart: Program to Binary

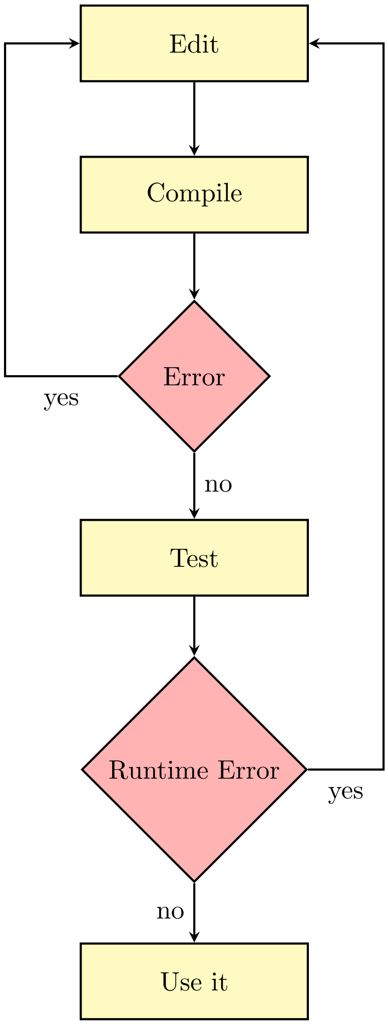
September 12, 2021

This is a flow chart of what is done by the computer when compiling a C++ file. In blue is the Python equivalent.



Note that the bottom of the flow chart is the same for all programming languages, because in all languages, CPU-specific machine code is needed to execute code.

The process of catching errors is as according to the following flow chart.



In this context, **errors** are caught by the compiler. This is opposed to **runtime errors**, which are not caught by the compiler. These can be something like division by zero or infinite loops.

3.2 First C++ Code

September 12, 2021

The following code is a hello world program in C++.

```
// helloworld.cpp
#include <iostream>  // include statement allows the use of C++ libraries
#include <stdio.h>   // this library contains getchar()

using namespace std; // a standard environment (input from keyboard, output is the screen)

int main() {

   cout << "Hello world!" << endl;  // Prints "Hello world!" to the screen

   getchar();    // wait for user to type a character

   return 0;    // 0 means that the execution was successful
}
```

3.3 Data types

September 12, 2021

Variables are referred to as identifiers. Identifiers are memory locations accessed and modified.

Inside a main function (as above), the following code declares a variable of integer type in C++.

```
int numYears;     // allocated space in memory to contain num of years
```

You can also initialize the variable with a value on declaration:

```
int number = 5;   // declare and initialize (give a value too)
```

Some rules to follow when naming variables are:

- Names have meanings
- Must be case sensitive (e.g. numYears is not numyears)
- Consists of letters, numbers and underscores
- First character cannot be a number

Some types of variables are:

- Integers (e.g. -5, 0, +2) [int]
- Real numbers (floating point numbers or doubles, e.g. 2.453, -4.1987e7) [float or double]
- Booleans (e.g. true, false) [bool]
- Characters [char]

You can assign a value to a variable after declaring it:

```
int width, height;
int area;

width = 5;
height = 3;

area = width * height;
```

Constants (denoted with the keyword "const") cannot be changed throughout the program. The convention is to use capital letters for constants.

```
const double PI = 3.14159265;

int radius = 6;
double area = PI * radius * radius;
```

You would get an error if you were to try to reassign a constant.

```
const double PI = 3.14159265;
PI = 3.14;  /// ERROR
```

3.4 Arithmetic Operations
September 22, 2021

There are 5 basic arithmetic operations supported by C++

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Modulus or "Remainder" (%)

For example,

```
int a = 5;
int b = 2;

cout << a + b / 2 << endl;  /// 6
cout << (a + b) / 2 << endl;  /// 3
cout << a / 2 + 3 << endl;  /// 5
cout << a / b / 3 << endl;  /// 0
```

This is done with integer operations since both arguments of each operation are integers (round down if the result of a division is a decimal). If one argument is a double/float, the result is a double/float. For example,

```
int a = 5;
int b = 2;

cout << a + b / 2 << endl;  /// 6
cout << (a + b) / 2 << endl;  /// 3.5
cout << a / 2 + 3 << endl;  /// 5
cout << a / b / 3 << endl;  /// 0.83333333
```

3.4.1 Division when initializing or assigning a variable

Naively, to initialize a variable to the result of a division, you could try

```
int div = 7/3;  /// 2
```

However, this results in 2, not the desired result of 2.3333. You could also try changing the variable type

```
double div = 7/3;  /// 2
```

This does not work either, since the integer division happens before the result is assigned to the variable div. Instead, you have to force the compiler to recognize an argument as a float.

```
double div = 7.0/3;   /// 2.333333
```

If instead you had decided to make the variable type of div and integer and run it

```
int div = 7.0/3;   /// 2
```

It will perform the double operation, but save it in var as an integer.

A way of doing the float operation with two integers is by **casting** types. This can be done as follows.

```
double div = (double) 7/3;  /// 2.333333
double div2 = 7/(double)3;  /// 2.333333

int c = 7
double div3 = (double) c/3;   /// 2.333333
// Note, this does not change the type of c.
```

The remainder (%) operation returns the remainder of the division between two integers:

```
cout << 7 % 3 << endl;   /// 1
cout << 10 % 6 << endl;  /// 4
cout << 13 % 4 << endl;  /// 1
```

3.4.2 Type conversion

We can also convert types from doubles to integers by truncation or rounding:

```
double price = 2.55;

int sum = price;   /// truncate
int sum2 = price + 0.5  /// round to the nearest int
```

3.5 I/Os

September 22, 2021

To do this, we will use the iostream library and standard namespace.

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

3.5.1 Reading inputs

You can get input from the keyboard using cin. For example,

```
int number;

cout << "Please enter a value between 0 and 10." << endl;

cin >> number;  // reads integer

cout << "The entered number is " << number;
```

You can also get multiple variables with the same cin.

```
int length, width;

cout << "Please enter the length and width of the rectangle: ";

cin >> length >> width;

int area = length * width;
cout << "The area is " << area
```