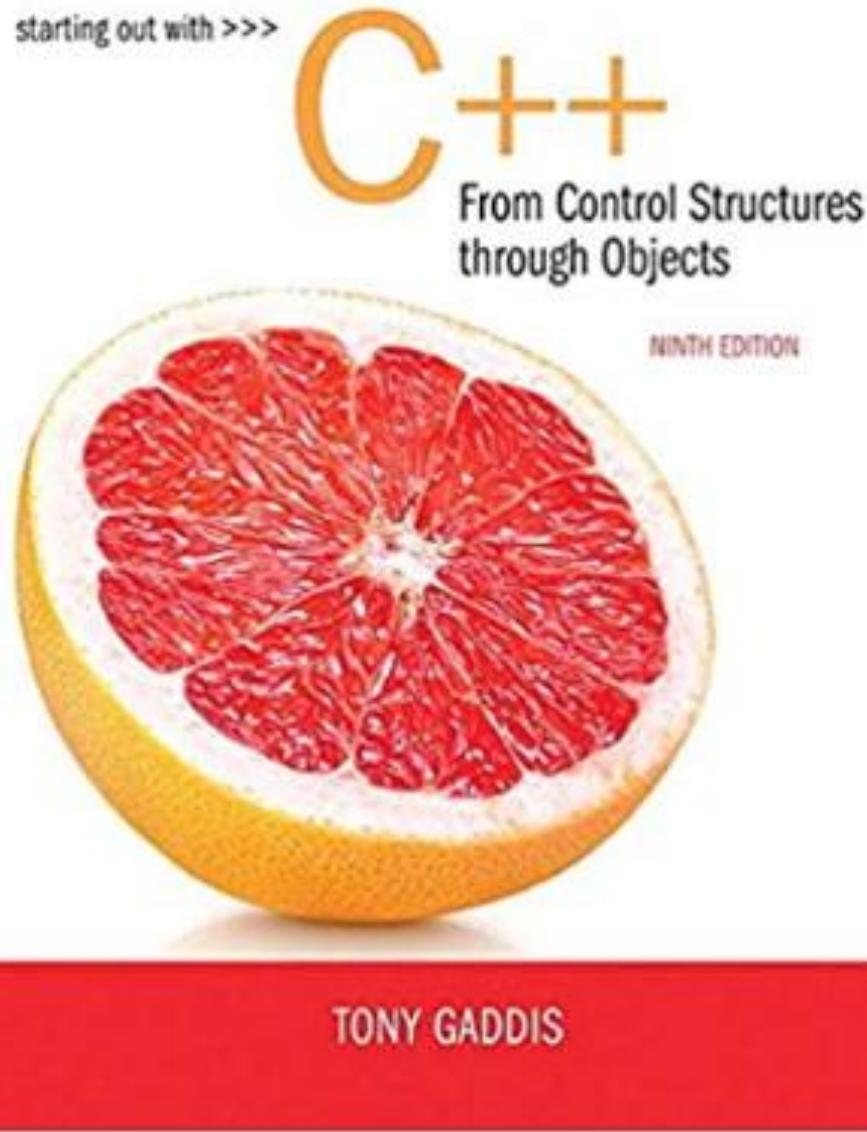


starting out with >>>



Textbook For CS-102-04
Can be bought or rented
as an eBook for the
semester at
Amazon.com for \$39.99

Introduction to Computers and Programming

Chapter 1

Prior to World War II, What was a Computer?

A Computer, before Digital Computers



A [NASA photo](#) of Katherine Johnson at work.

The 2016 movie: “Hidden Figures”, was about Katherine Johnson

The Analog Computer



- Vannevar Bush pushed the Analog Computer to the limits of what it could do.
- Being mostly mechanical, it was slow.
- The accuracy of its results was only 3 decimal places.

What makes a Digital Computer Different
from any Machine that had preceded it?

What makes a Computer Different from any Machine that had preceded it?

- It is a General Purpose Machine.

What makes a Computer Different from any Machine that had preceded it?

- It is a General Purpose Machine.
 - Limited only by the input device or output device, it will do whatever you tell it to do.

What makes a Computer Different from any Machine that had preceded it?

- It is a General Purpose Machine.
 - Limited only by the input device or output device, it will do whatever you tell it to do.
 - It will run any application that you care to put on it.

What makes a Computer Different from any Machine that had preceded it?

- It is a General Purpose Machine.
 - Limited only by the input device or output device, it will do whatever you tell it to do.
 - It will run any application that you care to put on it.
 - In the history of the Industrial revolution, nothing like it had ever occurred before.

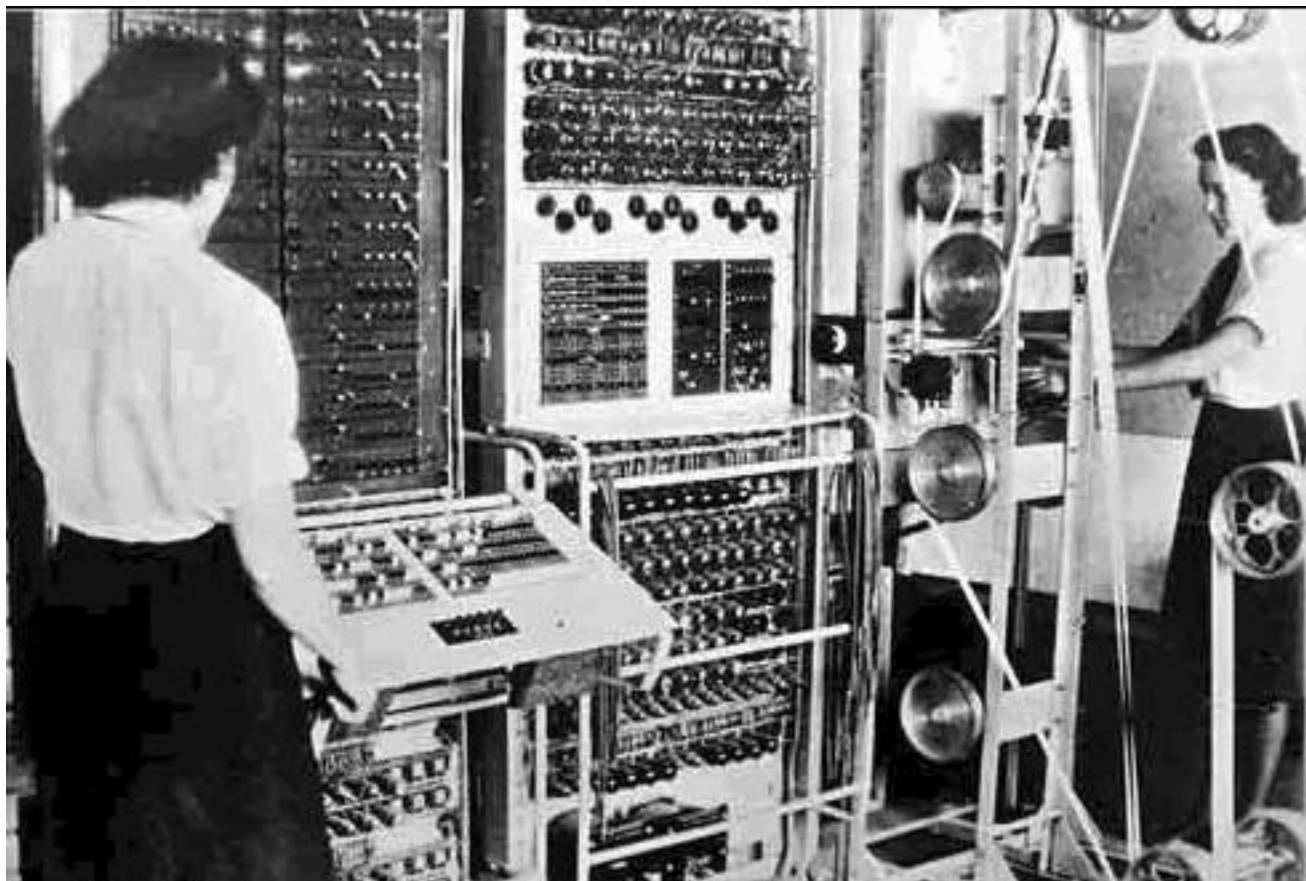


What makes a Digital Computer Different from any Machine that had preceded it?

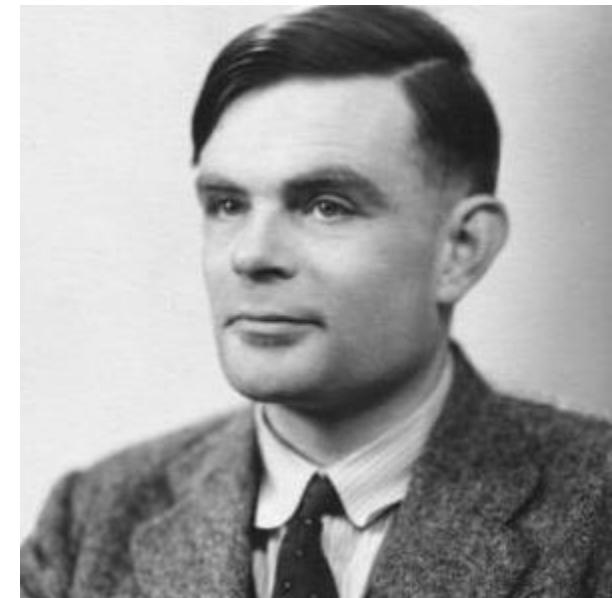
- It is a General Purpose Machine.
 - Limited only by the input device or output device, it will do whatever you tell it to do.
 - It will run any application that you care to put on it.
 - In the history of the Industrial revolution, nothing like it had ever occurred before.
- A digital computer is programmable.

The Colossus Mark 1 was the first Programmable, Electronic Digital Computer

- It was built by the British in Bletchley Park in 1943
- Designed by Tommy Flowers & inspired by Alan Turing



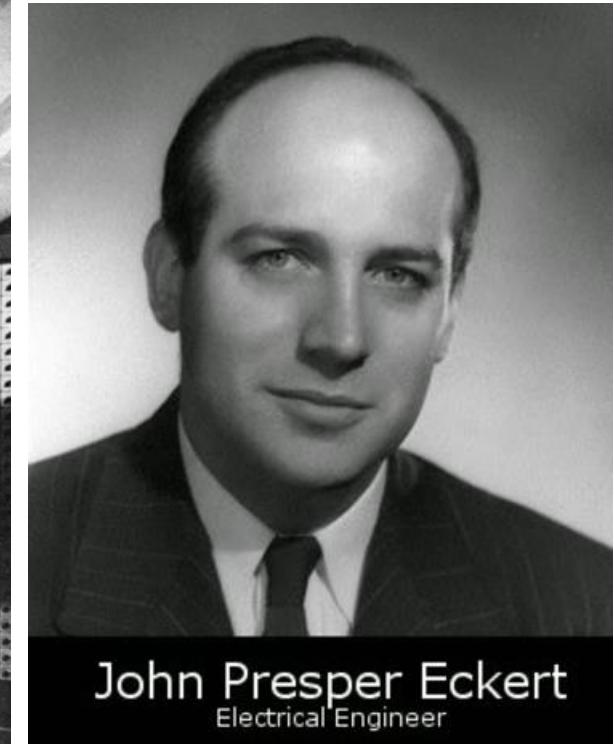
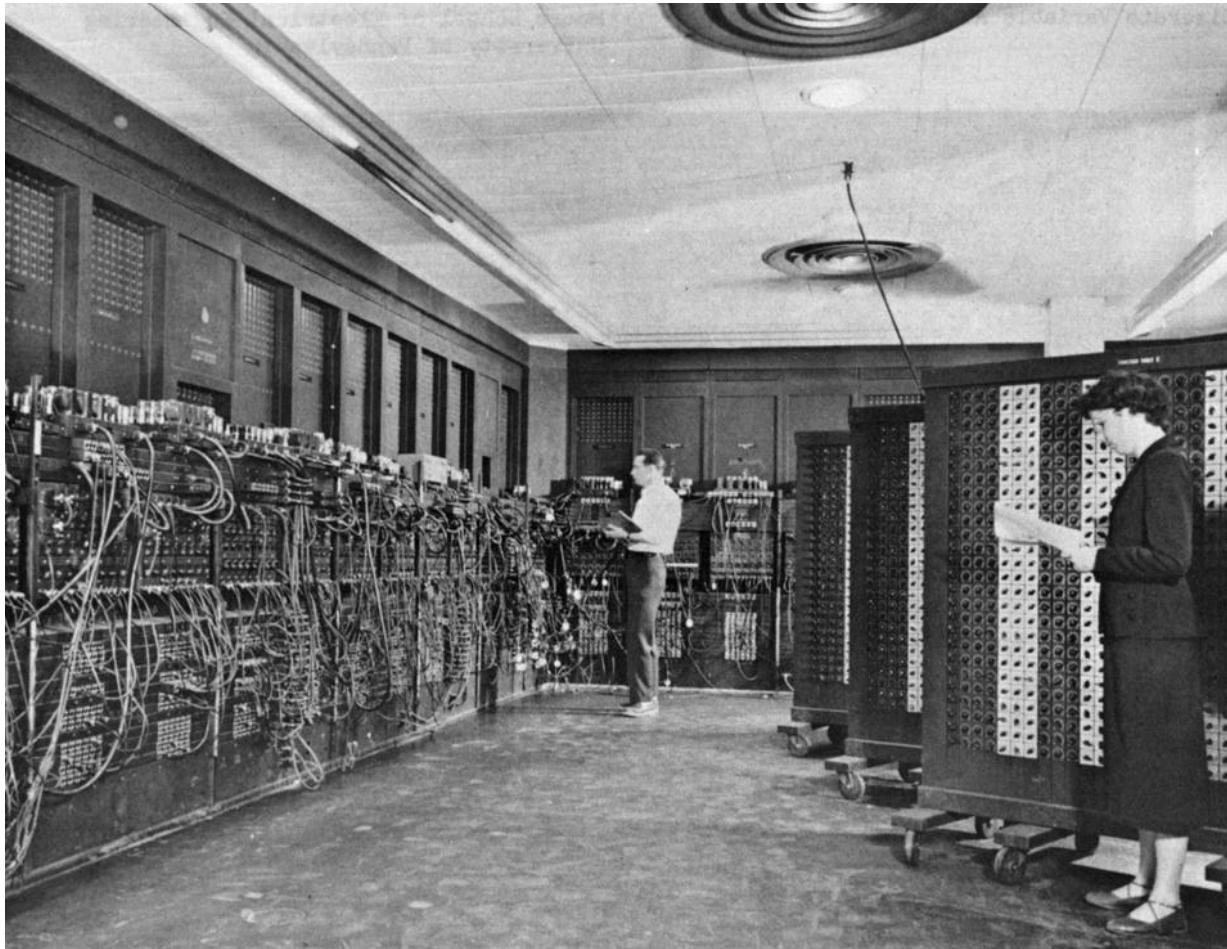
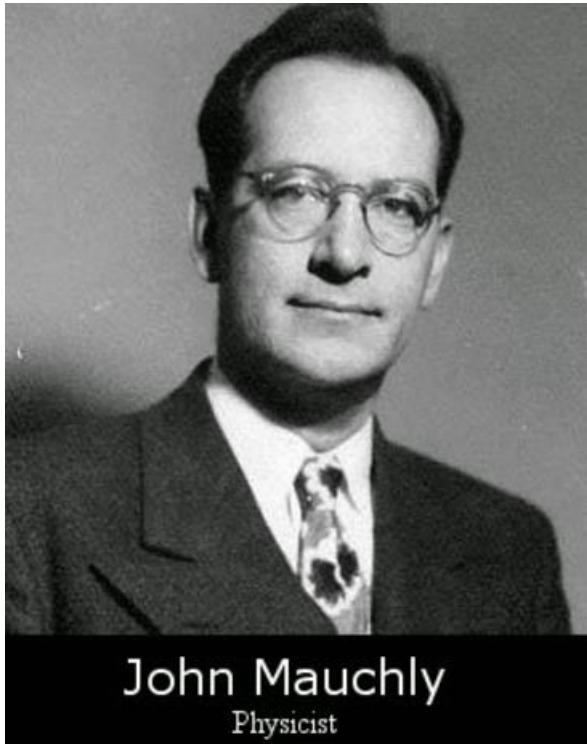
The Enigma



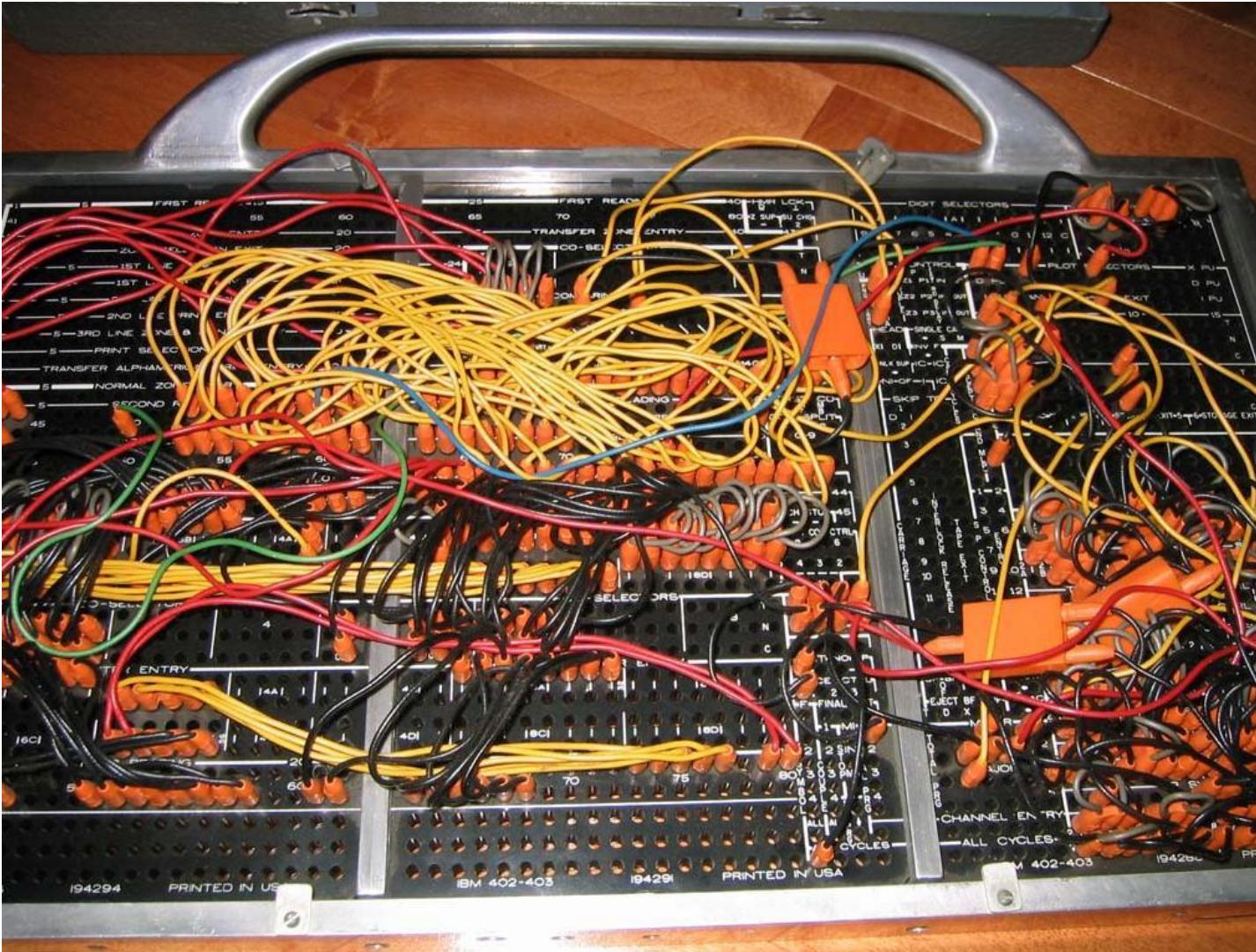
"The Imitation Game"

Eniac – The 1st Programmable General Purpose Electronic Computer (1945)

The Electronic Numerical Integrator And Computer was designed at the University of Pennsylvania by J. Presper Eckert and John Mauchly



It was programmed by Plugboards



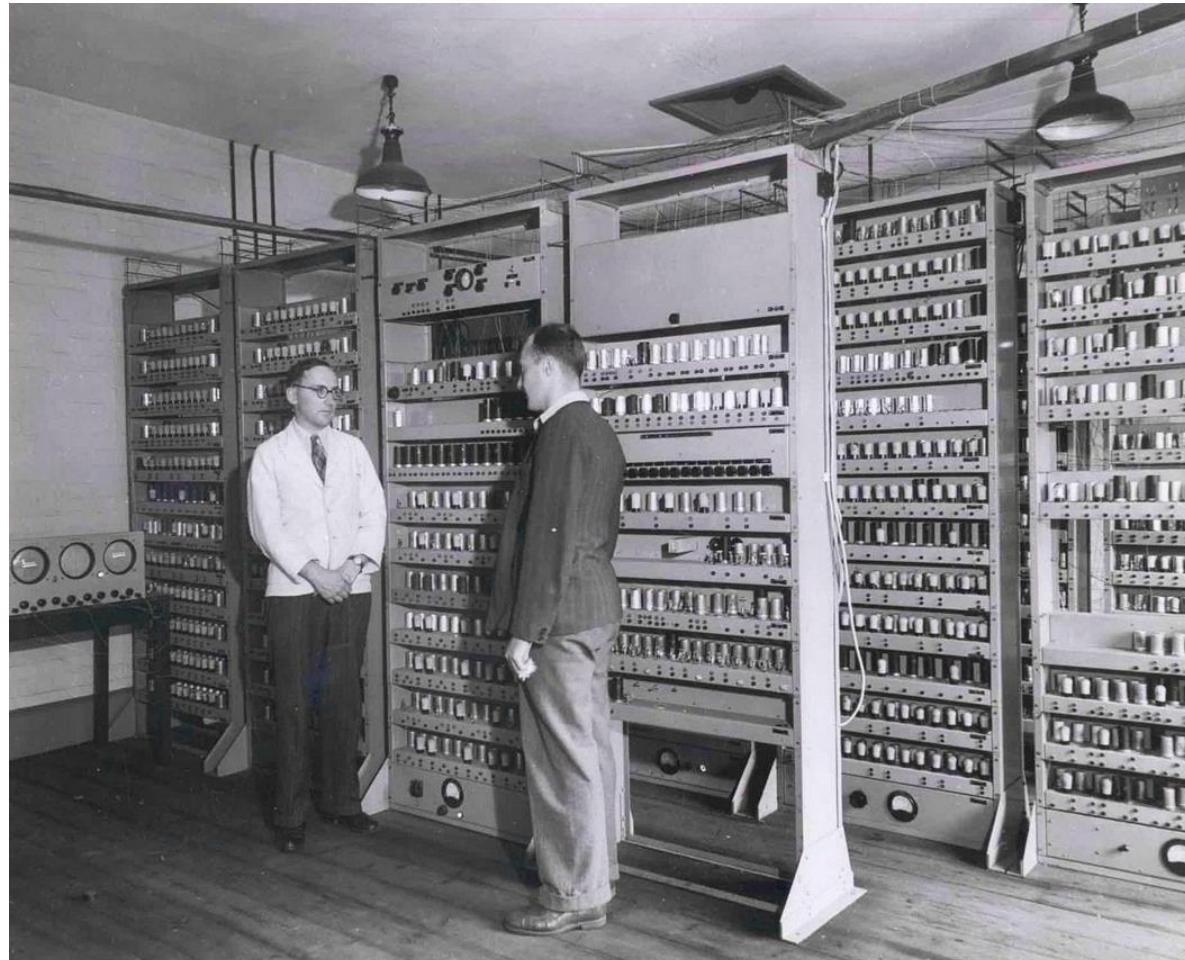


The Stored Program Concept

- John von Neumann, who consulted with Mauchly and Eckhart on the ENIAC wrote a seminal paper in which he proposed storing the program in memory along with the Data.
- From this point forward, all computers were designed using the “Stored Program Concept.”

EDSAC - the first complete and fully operational electronic digital Stored-Program computer: 1949

- Electronic Delay Storage Automatic Calculator - Maurice Wilkes



The Univac was the first Commercially Available Computer – Cost: \$1,000,000+

Built in 1951, by J. Presper Eckert and John Mauchly, it had a stored program memory.



TRADIC - The First Fully TRAnsistorized Digital Computer

- In 1955 Jean Howard Felker of Bell Labs built the first totally transistorized computer for use by the Air Force.



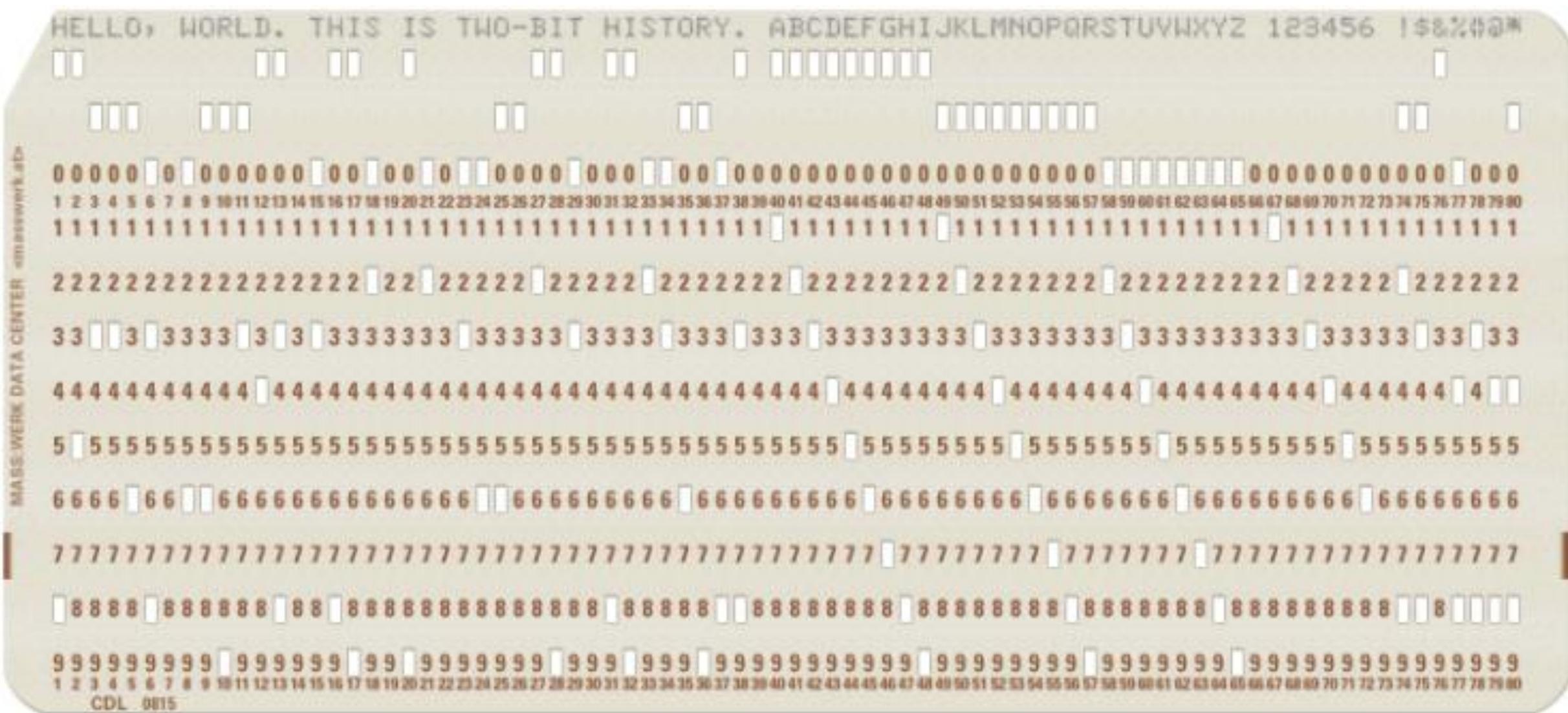
The IBM 7090, built in 1959, Defined what was to be called the MainFrame – Cost: \$2,900,000



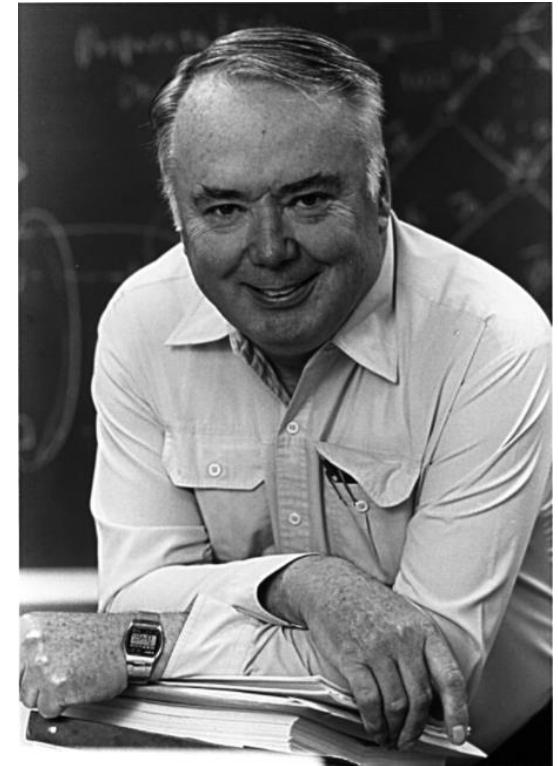
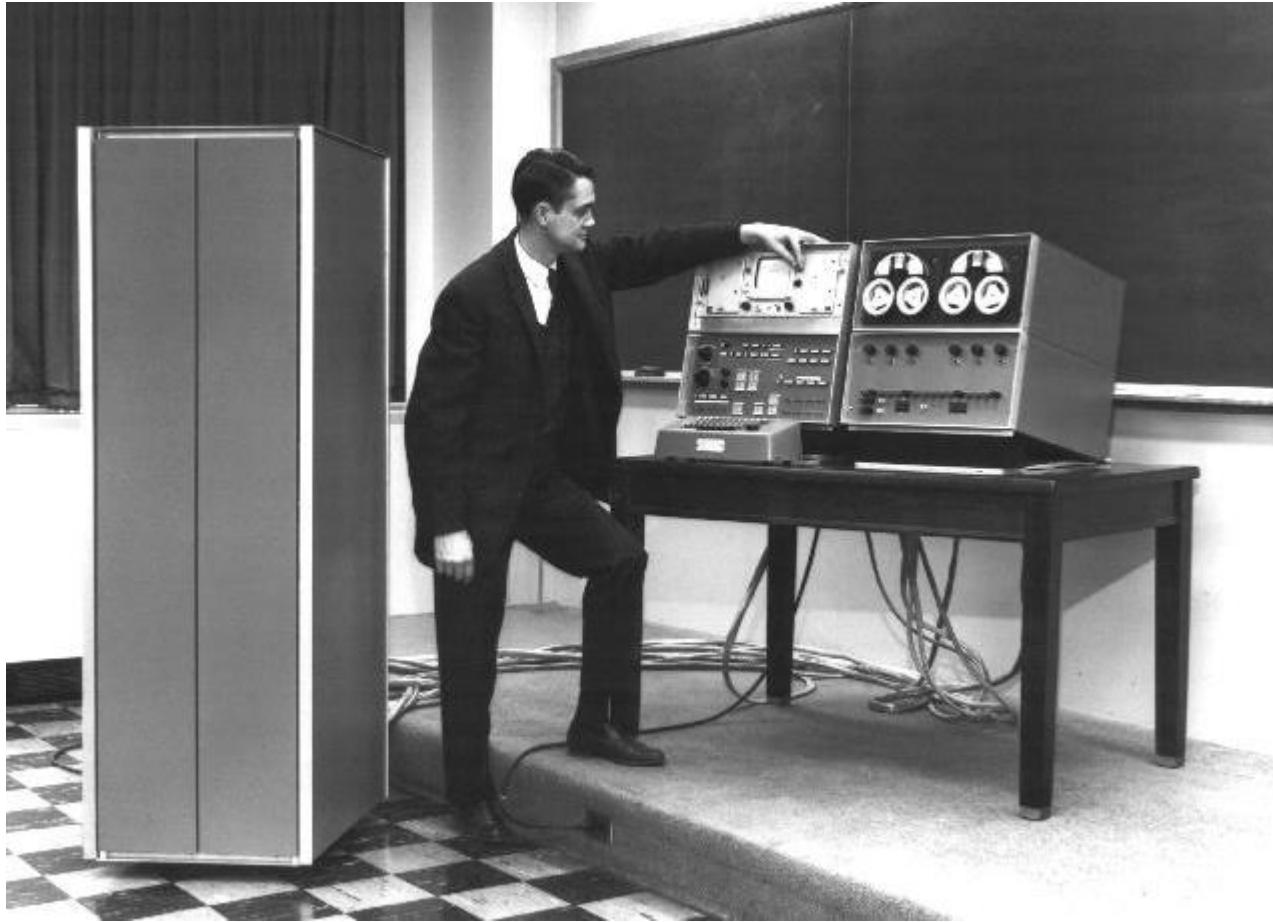
The IBM 7090 was the Transistor version of the tube IBM 709.

The Punched Card

How programmers related to Computers 1950's – 1960's



The first Minicomputer, The LINC, was built in 1962 and cost \$43,000 to build



The Laboratory INstrument Computer was designed and built by Wesley A. Clark and Charles E. Molnar of MIT Lincoln Labs

The Xerox PARC Alto – the 1st GUI

- The first Graphic User Interface was developed at Xerox PARC in Palo Alto in 1973 – Cost \$10,000

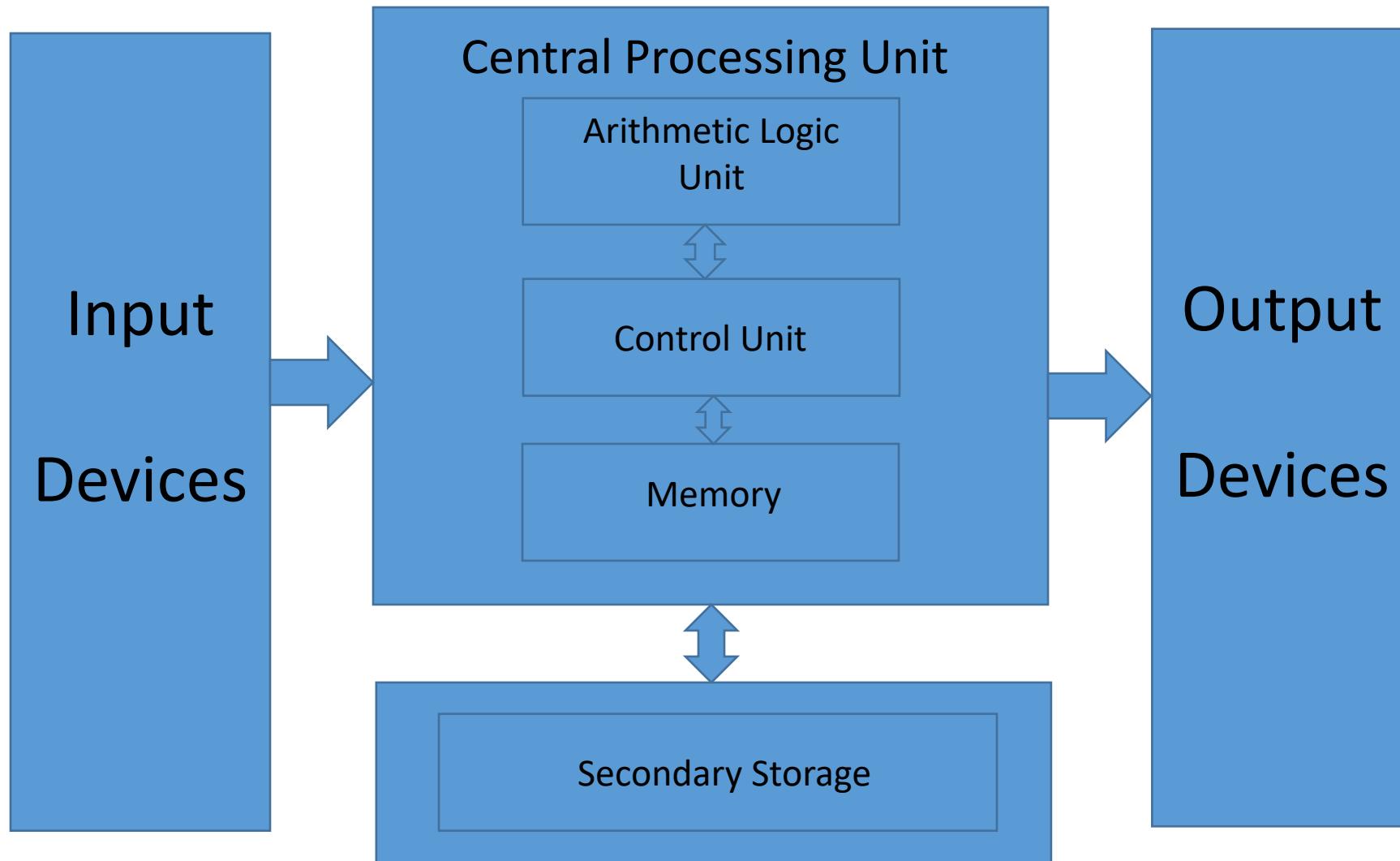


MITS Altair 8800 – 1st Microcomputer

- Introduced in 1975 as a Kit, it not only launched the Microcomputer Revolution but Microsoft as well. Cost: \$250



What All These Machines Have In Common:



Main Hardware Component Categories

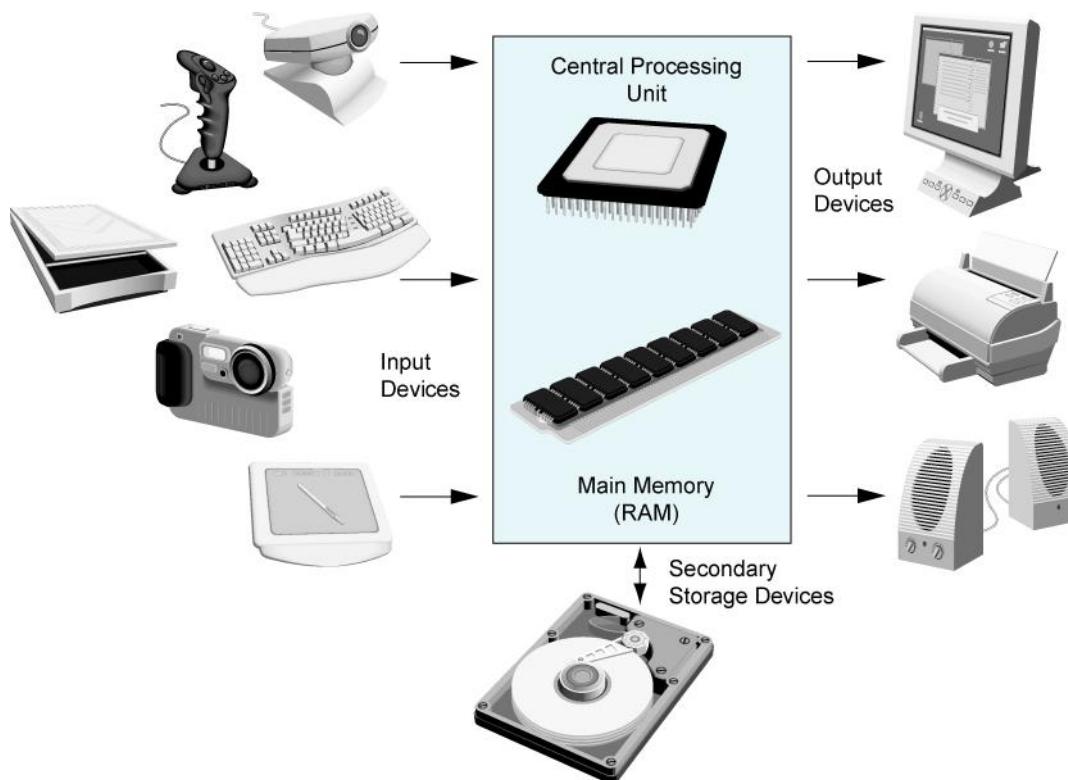


Figure 1-2

Main Memory

- It is volatile. Main memory is erased when program terminates or computer is turned off
- Also called Random Access Memory (RAM)
- Organized as follows:
 - bit: smallest piece of memory. Has values 0 (off, false) or 1 (on, true)
 - byte: 8 consecutive bits. Bytes have addresses.
- Addresses – Each byte in memory is identified by a unique number known as an *address*.

Main Memory

0		1		2		3		4		5		6		7		8		9
10		11		12		13		14		15		16	149	17		18		19
20		21		22		23	72	24		25		26		27		28		29

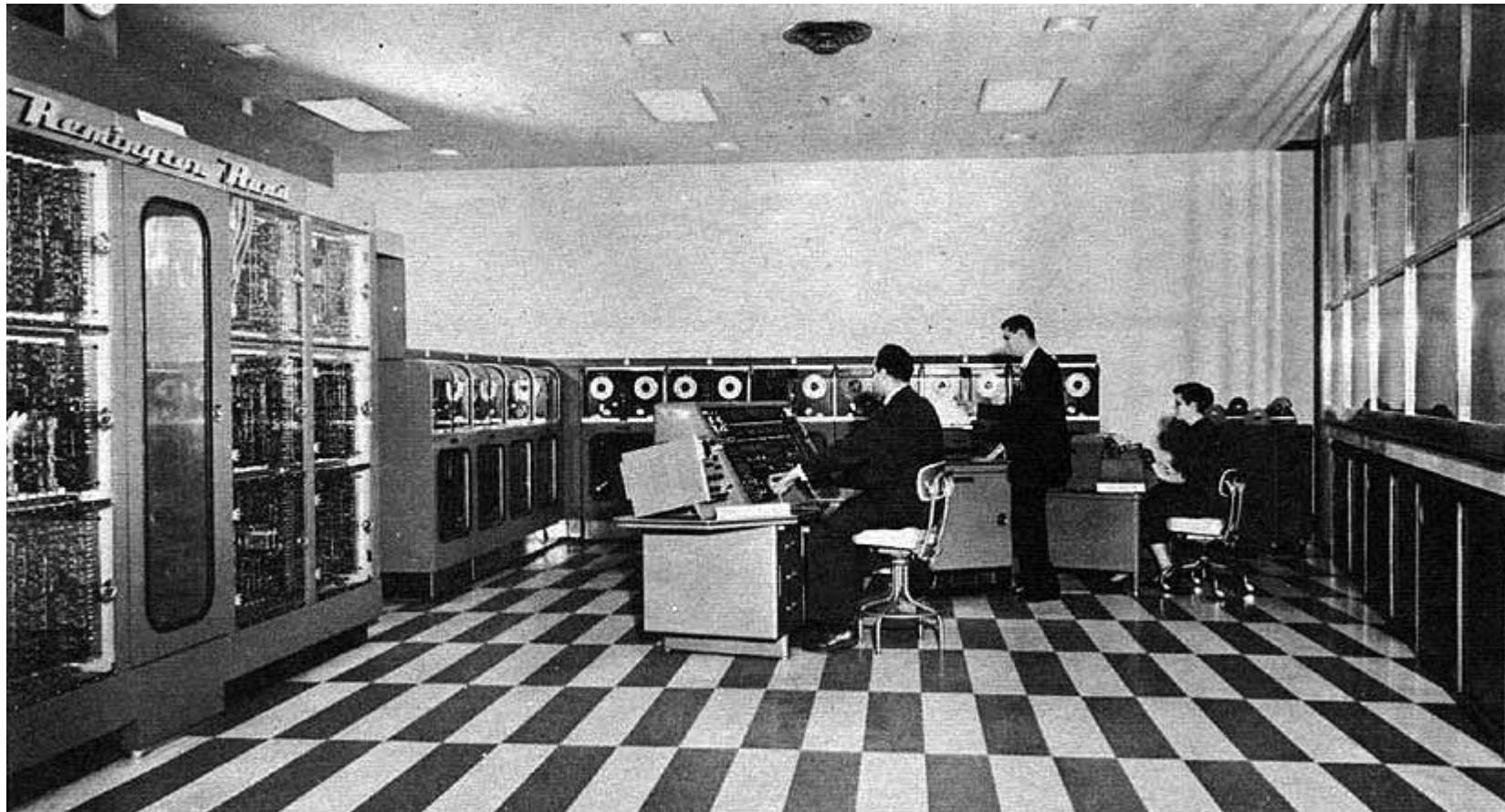
- Above, the number 149 is stored in the byte with the address 16, and the number 72 is stored at address 23.

Binary To Decimal Conversion

Software

The Univac was the first Commercially Available Computer – Cost: \$1,000,000+

The Univac was programmed in Machine Language



Example of Machine Language

First Generation Language

0020	1020
0021	0020
0022	0004
0023	0061
0024	0367
0025	0062
0026	7676
0027	6056
0030	0460
0031	6042
0032	0500
4033	6031
0034	6030
0035	0500
4036	6036
0037	1063
0040	0224
0041	6030
0042	1000
0043	0001
0044	1560
0045	7000
0046	1620
4047	1000
0050	4052
0051	0714
0052	0000
0053	0440
0054	6027
0055	6015
0056	0221
0057	0063
0060	4377
0061	0064
0062	7377
0063	0222
0064	6000
0065	1020
0066	7777
0067	0000



The first Minicomputer, The LINC, was programmed strictly in Assembly Language. The author of its Operating System was: Mary Allen Wilkes She did all her early programming from home on what was to be the first “Personal Computer”.

Example of Assembly Language

Second Generation Language

```
START, LDA I          /I-O PRESET CODE
      20             /DO I-O PRESET
      ESF            /INITIALIZE POINTER
      SET I BLOCKN   /TO WORKING AREA=1
      WKAREA=1       /SET COUNTER TO WORKING
      SET I BLOCKC   /=WKSIZEx1 TO ALLOW
      =WKSIZEx1      /FOR RESET
      /RESET
      /LOOP
      /INLOOP, SNS I Ø           /SENSE SWITCH Ø NOT UP?
      /JMP DUMP          /IT IS UP, DUMP BUFFER
      /IOB
      /PMODE            /DO PDP-8 IOT
      /KSF              /IS A CHARACTER READY?
      /LMODE            /NO CHAR THERE, WAIT
      /JMP INLOOP        /CHAR THERE
      /IOB
      /PMODE            /READ IN THE CHAR
      /KRB              /PLACE IN BUFFER
      /LMODE            /BCOUNT
      /STA I POINT     /BUFFER FULL?
      /XSK I BCOUNT   /NO, GET ANOTHER WORD.
      /JMP INLOOP        /YES, WRITE OUT BUFFER
      /DUMP, LDA I      /BLOCKN
      /BLOCKN           /ONLY 9 BITS OF INTEREST
      /BCL I            /AND WITH 777
      /=777              /SET CORRECT FIELD BITS
      /BSE I            /FOR 12 BIT CALCULATION
      /PMODE            /BUFFER+400&3000 /SETS CORRECT BITS ON
      /LMODE            /STC .+2 /GET BLOCKN NUMBER AND PLACE
      /                  /IN WRITE INSTRUCTION
      /WRC U            /WRITE OUT ON UNIT 1
      /0000              /BLOCK NUMBER GOES HERE
      /SNS Ø            /SENSE SWITCH Ø UP?
      /JMP LOOP          /GO BACK AND GET SOME MORE
      /JMP DIAL          /YES, LOAD IN DIAL.
      /
      /RESET, XSK I BLOCKN /INCREMENT BLOCK NUMBER
      /SET I POINT      /SET BUFFER POINTER
      /BUFFER=1
      /SET I BCOUNT    /SET BUFFER WORD COUNTER NOW
      /=400              /400 OCTAL=256 DECIMAL
      /XSK I BLOCKC   /TOO MUCH DATA?
      /JMP Ø            /NO, RETURN
      /LDA I 7777        /7777
      /HLT              /YES, HALT WITH 7777 IN AC
```

MainFrames, like the IBM 7090 could run high level (3rd Generation) languages, as well as Assembly Languages

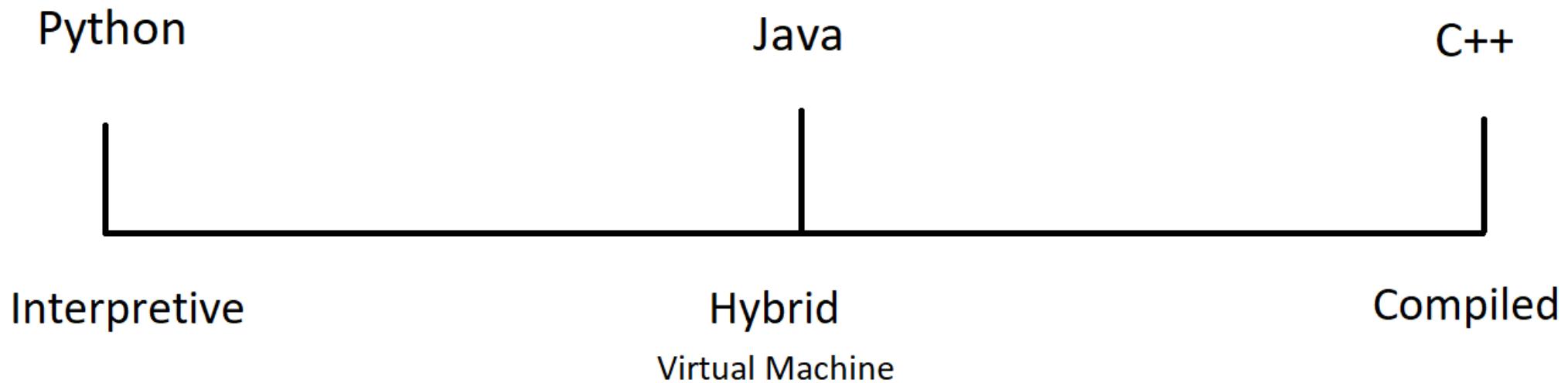


Third Generation Languages quickly became the favorite means to Program.

A Short List of 3rd Generation Languages

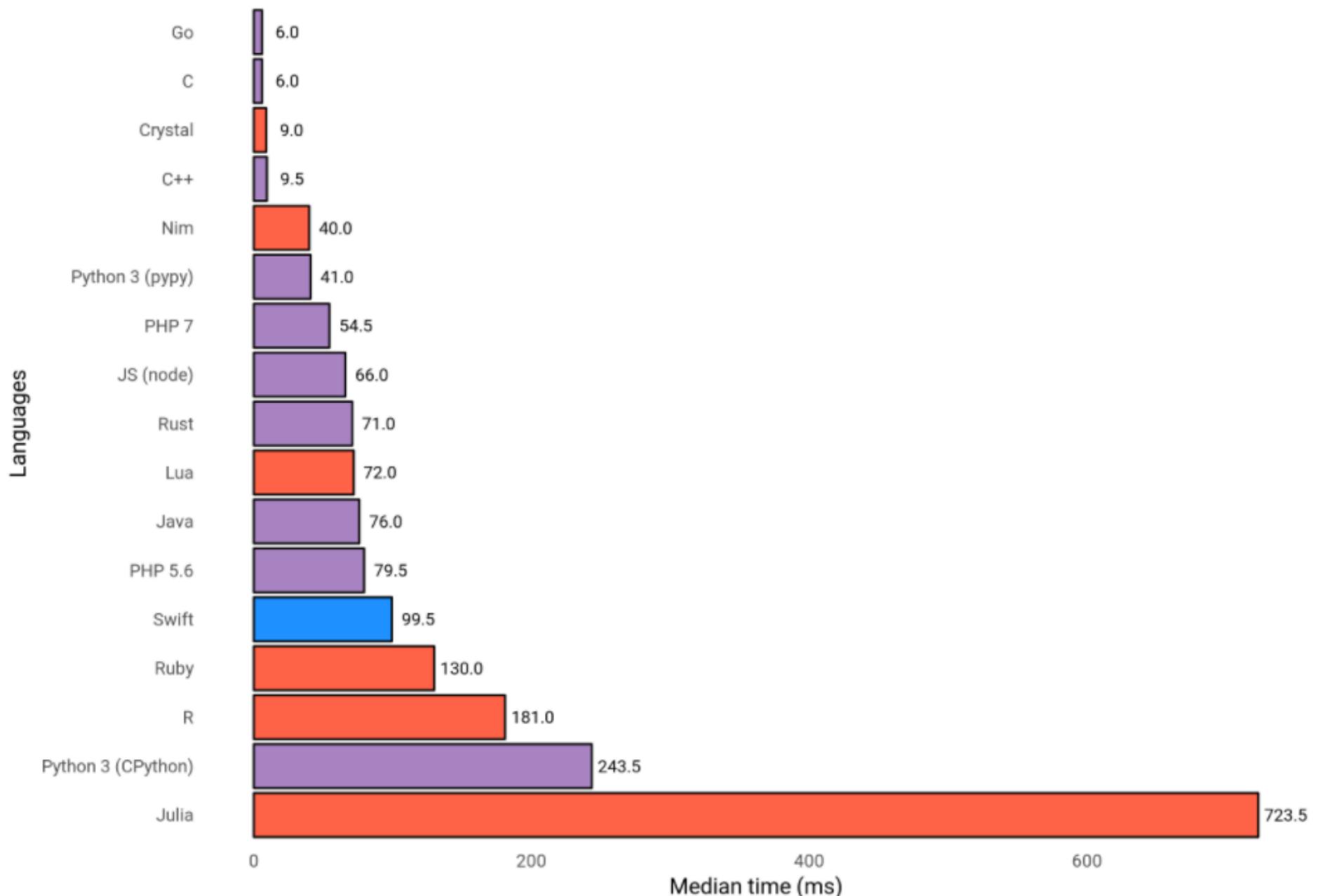
- FORTRAN – Formula Translator (1957)
- COBOL – Common Business Oriented Language (1960)
- BASIC – Beginners All-purpose Symbolic Instruction Code (1964)
- Pascal – Structured language used for teaching coding (1970)
- C – Structured language built by Bell labs (1973)
- C++ – Offers Object Oriented features not found in C (1983)
- Python – High Level Interpreter Object Oriented language (1989)
- Visual Basic – Designed to create Windows based apps (1991)
- Java – Object Oriented designed to run on multiple platforms (1995)
- C# – Designed by Microsoft for use on Web servers (2000)

Spectrum of 3rd Generation Languages



Speed comparison of various porgramming languages

Method: calculating π through the Leibniz formula x times



Why C++

- There are lots of programming languages out there, why are we paying attention to C++?
- What makes C++ so special, anyway?

Comparing C++ to QuickBasic

- Here is the most common “first program” that people often write, written in Quick Basic:

Print “Hello World”

End

- Below is the output this program produces:



The image shows a screenshot of a terminal window. At the top, there is a path: "C:\Classes\Class01\FirstProgram". Below the path, the text "Hello World" is displayed in white on a black background. The "Hello" part is in blue and the "World" part is in red.

Comparing C++ to QuickBasic

- Here is the most common “first program” that people often write, written in C++

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World";
    return 0;
}
```

- This prints out the same thing but is 7 lines long.

Why C++

- Any features that you do not use you do not pay for (e.g. in performance).
 - The QB program creates an executable which is 653 Kbytes in size.
 - The C++ program creates an executable which is 464 Kbytes in size.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World";
    return 0;
}
```

- The header, `#include <iostream>` line, tells the compiler what parts of the compiler to add to the source code in order to run it.
- In this way, no unnecessary code is placed in the executable file which is created by the compiler.

Why C++

- In this course, every program that we will study is considered a short program and will likely not exceed more than a couple of pages in length.
 - These could all be written in QuickBasic, easily.
- However, C++ is made to make writing very large programs, written by more than one person (a team perhaps), easy to put together in a way that one person's coding style and naming conventions doesn't affect another person's style.
- Microsoft Office, for example, has millions of lines of code and is written in C++.
 - It could only be written by a team of people.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World";
    return 0;
}
```

- The “using namespace std;” defines the part of the program where you write in the style that you like using the variable names that you want to use without worrying that your naming conventions will affect anyone else.
- In this way a team of programmers can work on a program and their individual contributions will not affect anyone else’s contribution.

The “Overhead” Helps

- So, even though there appears to be “unnecessary overhead” in every program, what you discover is that it has a definite purpose which will help you in the long run.
- So, it is to our benefit that we learn to program in a language which is scalable upwards, yet compact and efficient.
- By using C++ we will learn good programming techniques that will help us when we write large programs or work in a team.

What you will learn in this course

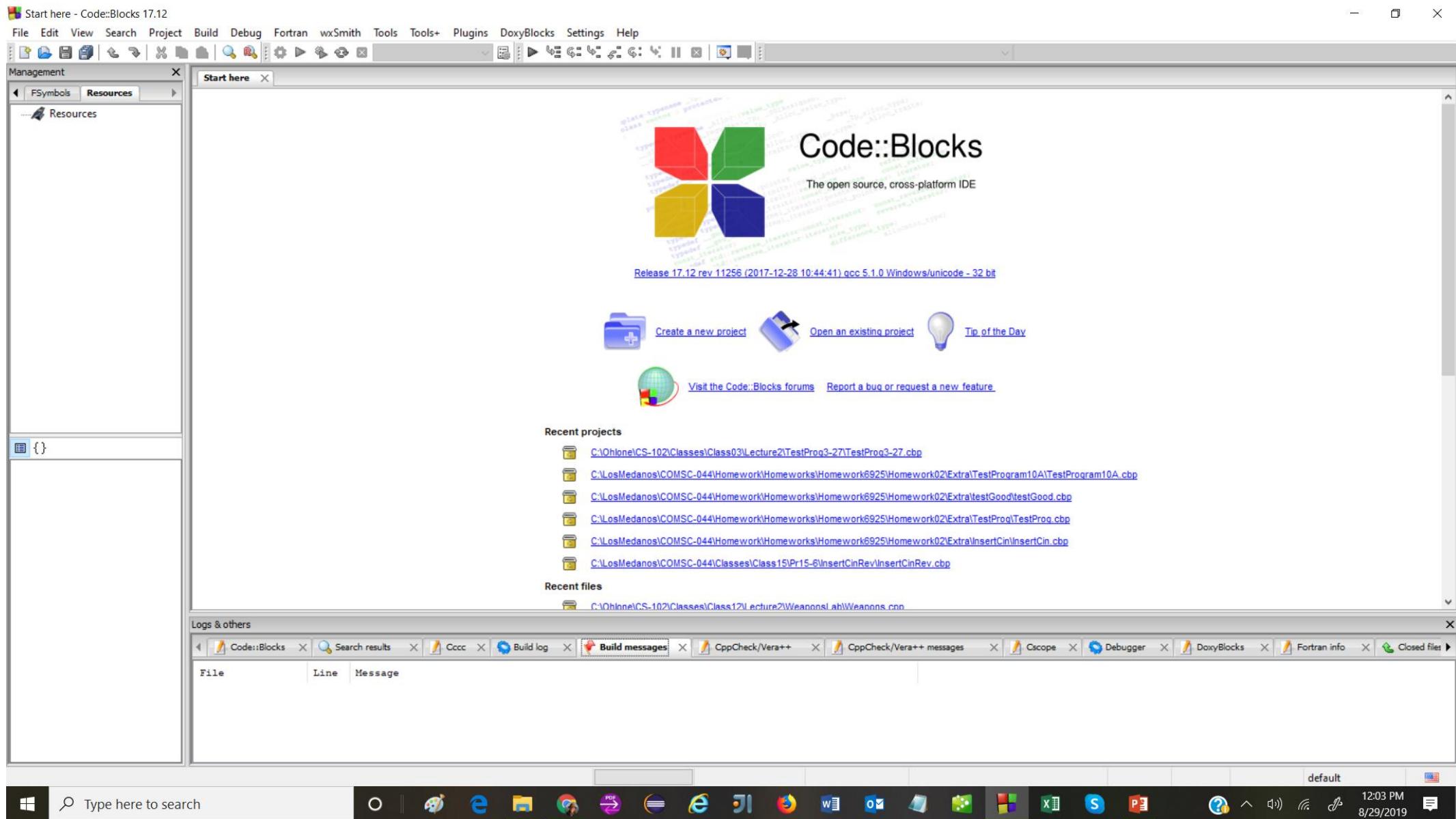
1. Data types and their uses.
2. Simple sequence programming
3. Use of selection (e.g. the **if** statement)
4. The loop (the **for** loop and the **while** loop)
5. Use of text files to save your data.
6. Use of functions (divide and conquer)
7. Use of Arrays and Vectors.
8. Sorting techniques (what a difference a good algorithm can make)
9. Use of Pointers to make processing more efficient
10. Programming Strings as well as numeric data
11. Using Data Structures
12. Use of binary and direct access files

Intro to CodeBlocks

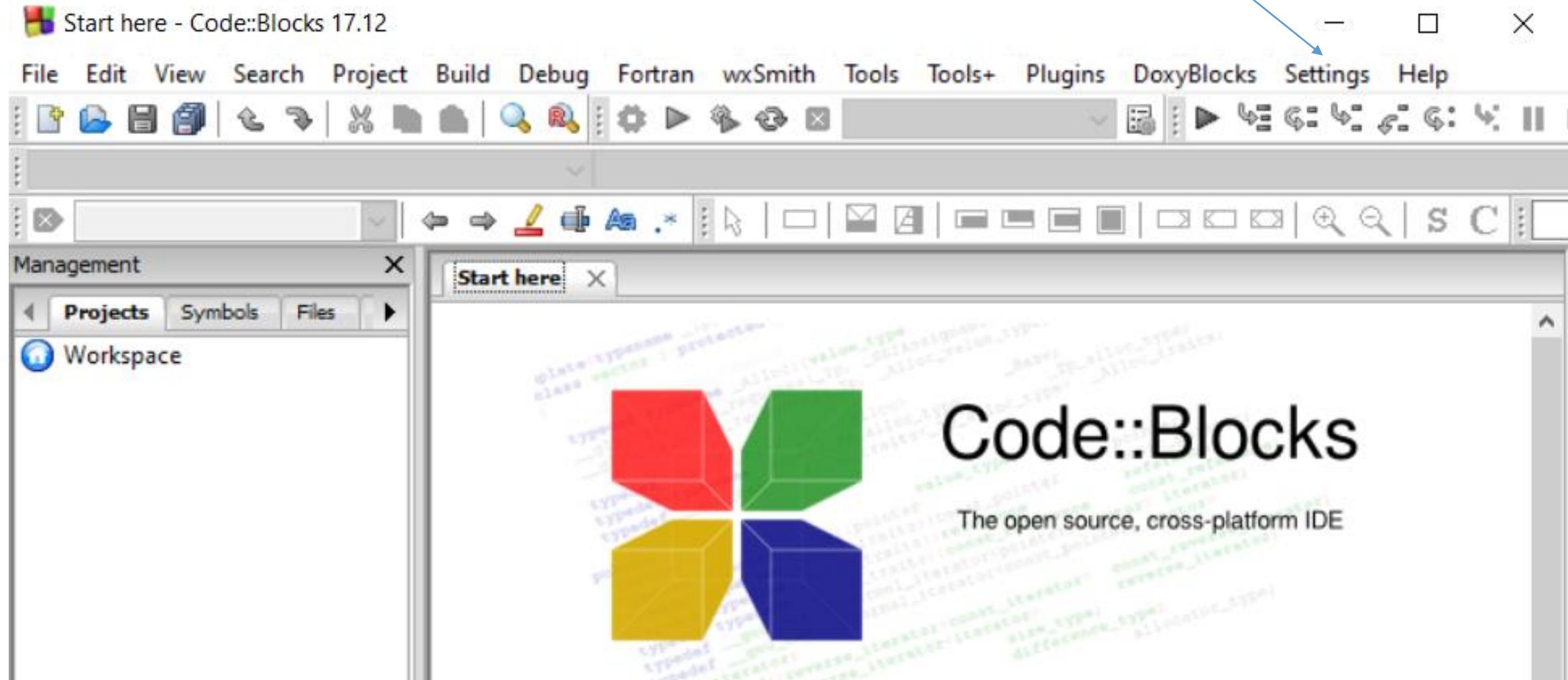
Mac Littlefield

CS-102

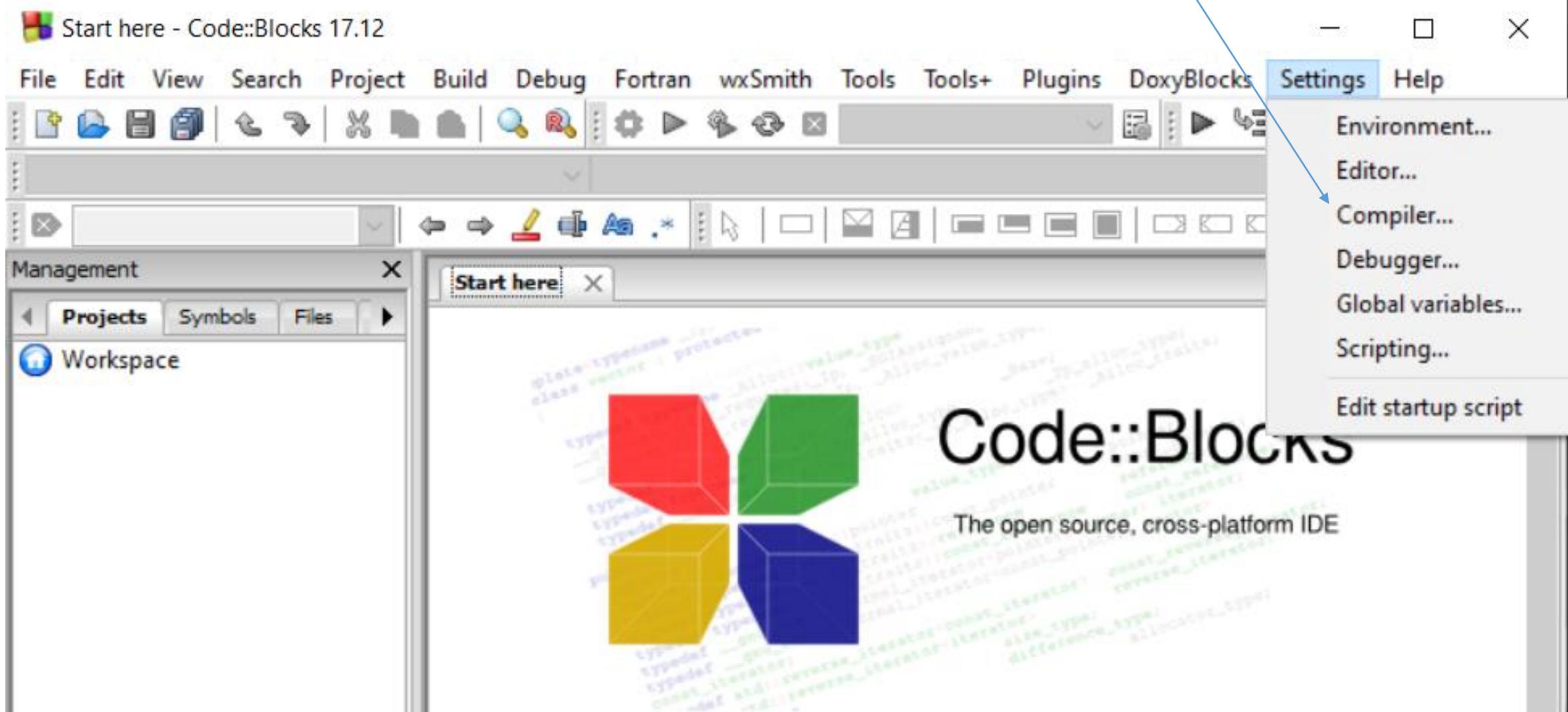
Opening Screen in CodeBlocks



Let's Click on Settings



Now Click on Compiler



You May find no compiler has yet been Selected

Compiler settings



Global compiler settings

Selected compiler: GNU GCC Compiler

Buttons: Set as default, Copy, Rename, Delete, Reset defaults

Tabs: Compiler settings, Linker settings, Search directories, Toolchain executables, Custom variables, Build options, Other settings

Policy: [dropdown menu]

Compiler Flags Tab:

- General
 - Have g++ follow the 1998 ISO C++ language standard [-std=c++98]
 - Have g++ follow the C++11 ISO C++ language standard [-std=c++11]
 - Have g++ follow the C++14 ISO C++ language standard [-std=c++14]
 - Have g++ follow the coming C++0x (aka c++11) ISO C++ language standard [-std=c++0x]
 - Have g++ follow the coming C++1y (aka C++14) ISO C++ language standard [-std=c++1y]
 - Have g++ follow the coming C++1z (aka C++17) ISO C++ language standard [-std=c++1z]
 - Have gcc follow the 1990 ISO C language standard (certain GNU extensions that aren't part of the standard) [-std=gnu90]
 - Have gcc follow the 1999 ISO C language standard [-std=c99]
 - Have gcc follow the 2011 ISO C language standard [-std=c11]
 - In C mode, this is equivalent to -std=c90, in C++ mode, it is equivalent to -std=c++0x [-std=c90]
 - Position Independent Code [-fPIC]
 - Static libgcc [-static-libgcc]
 - Static libstdc++ [-static-libstdc++1]

NOTE: Right-click to setup or edit compiler flags.

Icons on left sidebar: Global compiler settings (gear), Profiler settings (camera), Batch builds (gear and stack).

Choose either C++11 or C++14

Compiler settings

Global compiler settings

Selected compiler: GNU GCC Compiler

Compiler settings Linker settings Search directories Toolchain executables Custom variables Build options Other settings

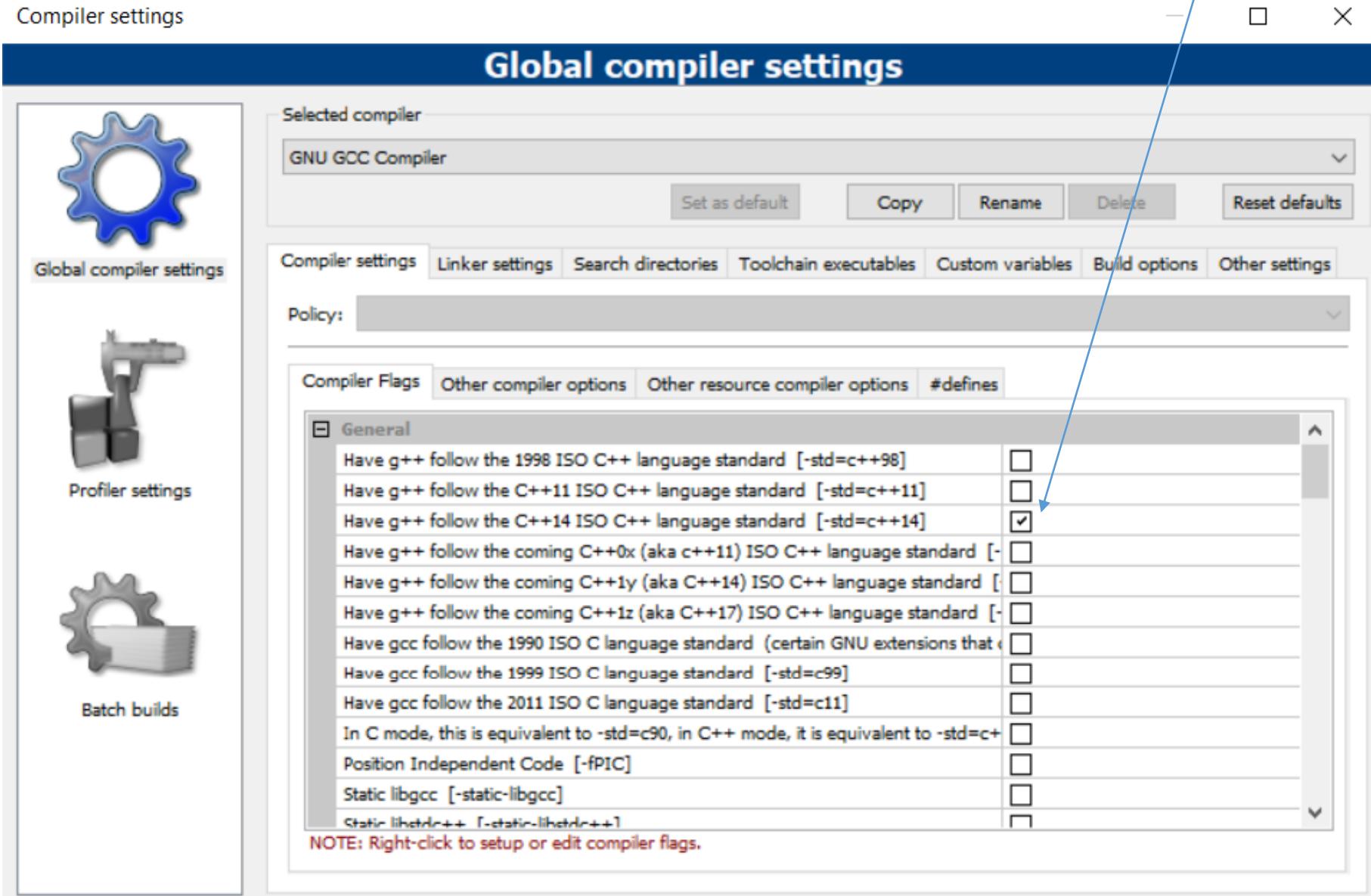
Policy:

Compiler Flags Other compiler options Other resource compiler options #defines

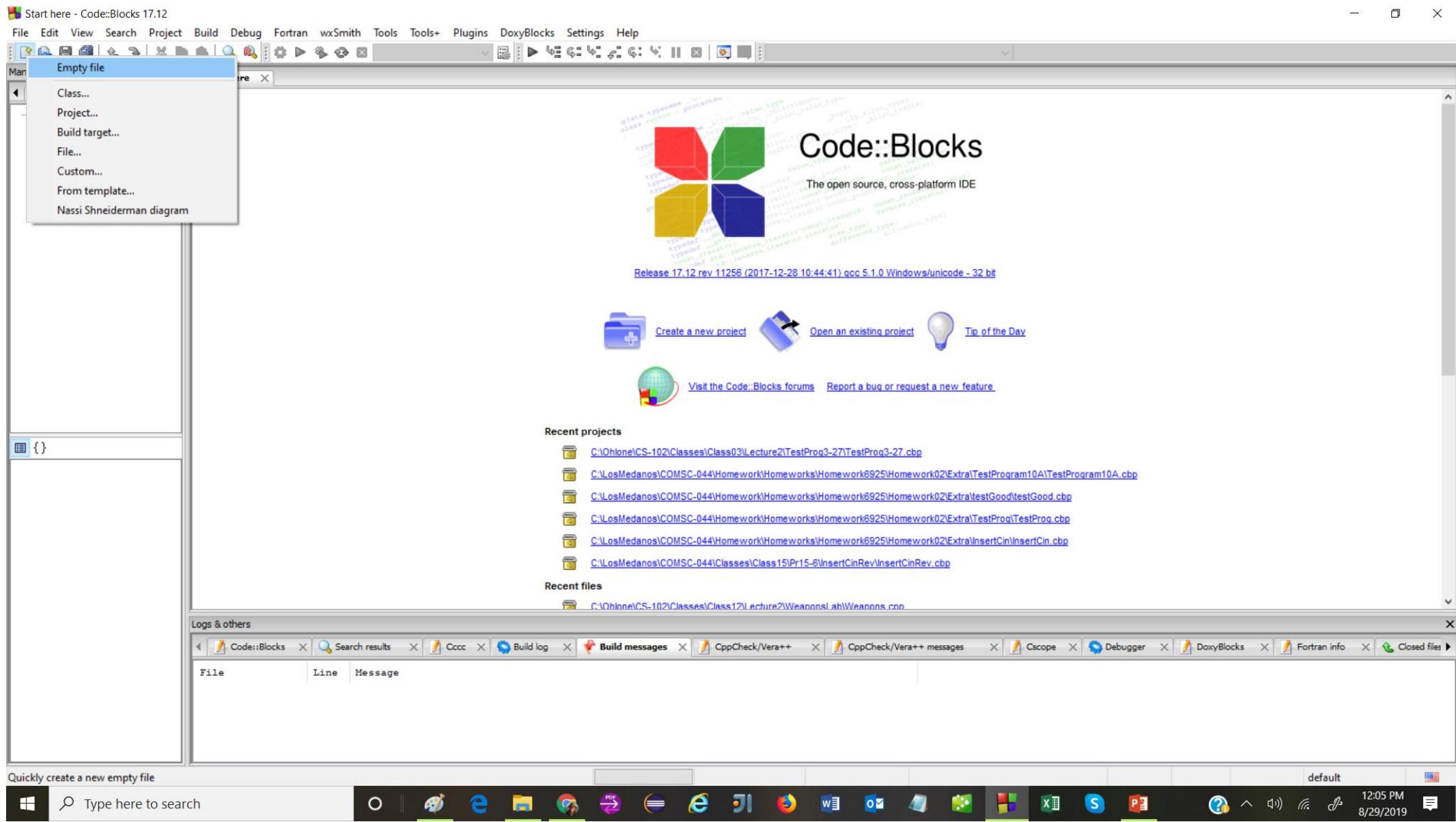
General

- Have g++ follow the 1998 ISO C++ language standard [-std=c++98]
- Have g++ follow the C++11 ISO C++ language standard [-std=c++11]
- Have g++ follow the C++14 ISO C++ language standard [-std=c++14]
- Have g++ follow the coming C++0x (aka c++11) ISO C++ language standard [-std=c++0x]
- Have g++ follow the coming C++1y (aka C++14) ISO C++ language standard [-std=c++1y]
- Have g++ follow the coming C++1z (aka C++17) ISO C++ language standard [-std=c++1z]
- Have gcc follow the 1990 ISO C language standard (certain GNU extensions that aren't part of the standard are disabled) [-std=c90]
- Have gcc follow the 1999 ISO C language standard [-std=c99]
- Have gcc follow the 2011 ISO C language standard [-std=c11]
- In C mode, this is equivalent to -std=c90, in C++ mode, it is equivalent to -std=c++0x [-std=c90]
- Position Independent Code [-fPIC]
- Static libgcc [-static-libgcc]
- Static libstdc++ [-static-libstdc++]

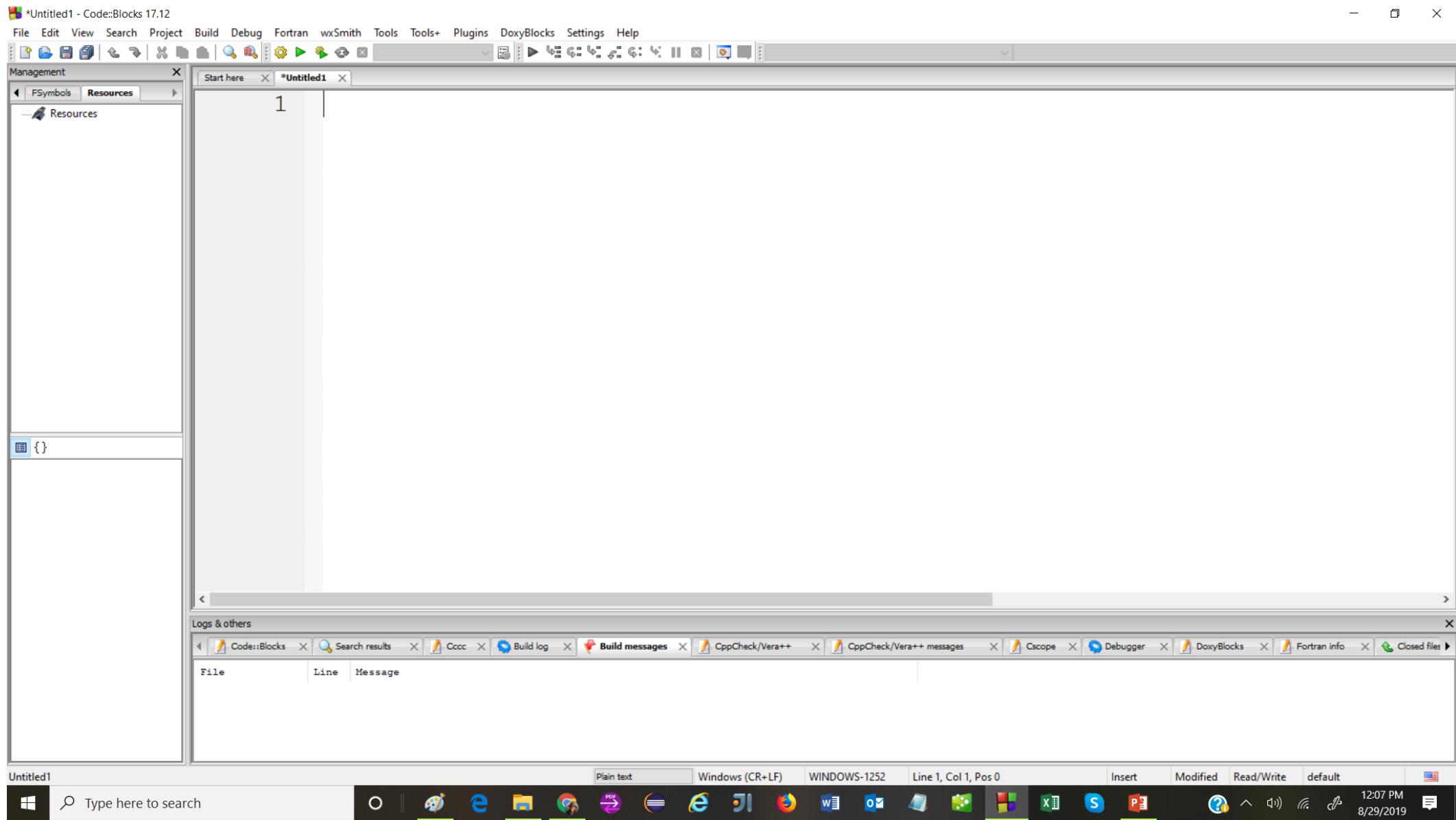
NOTE: Right-click to setup or edit compiler flags.



Select Empty File



Default New File Screen



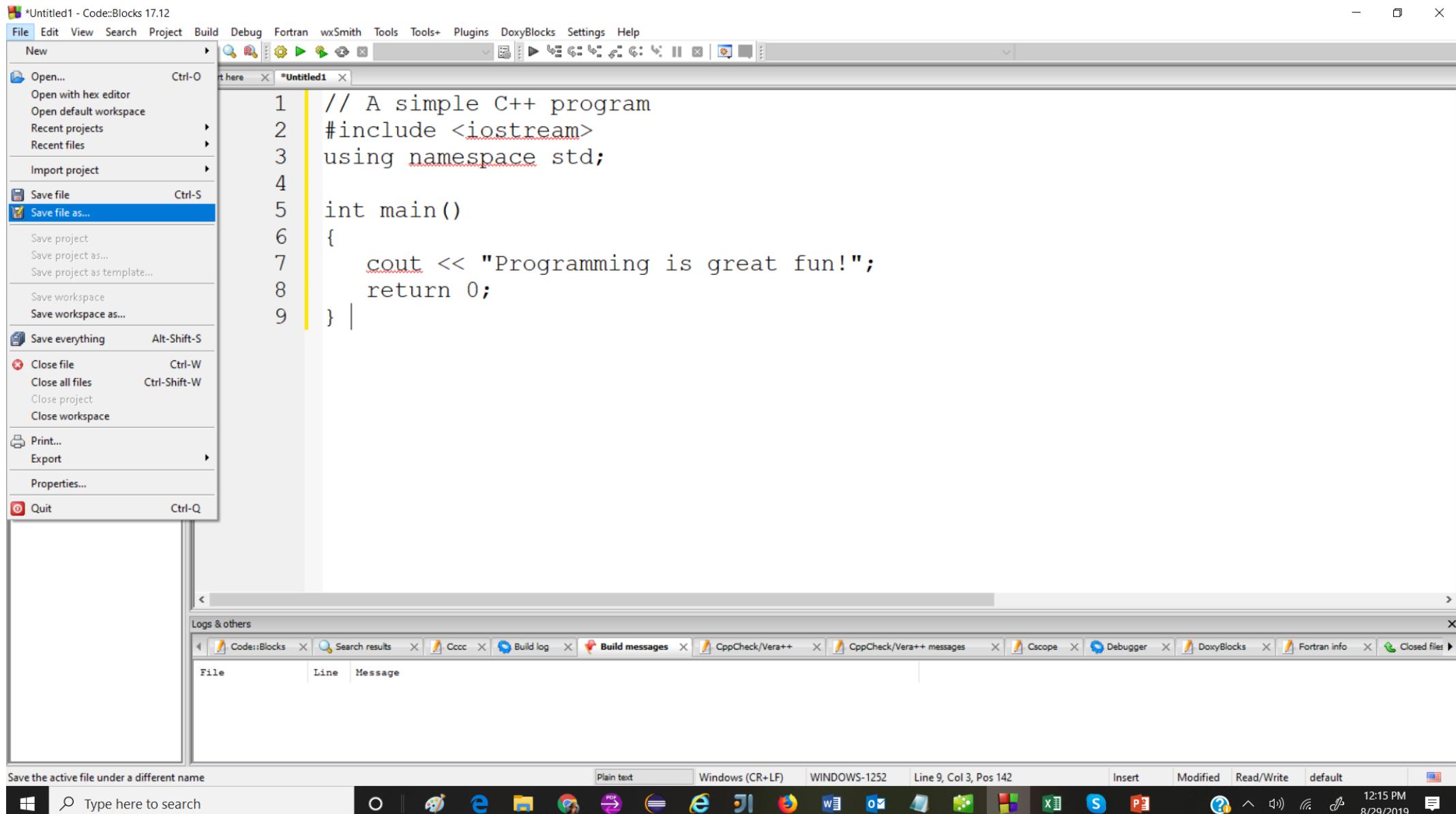
Typing in a Simple Program into Default File

The screenshot shows the Code::Blocks IDE interface. The main window displays a code editor with the following C++ code:

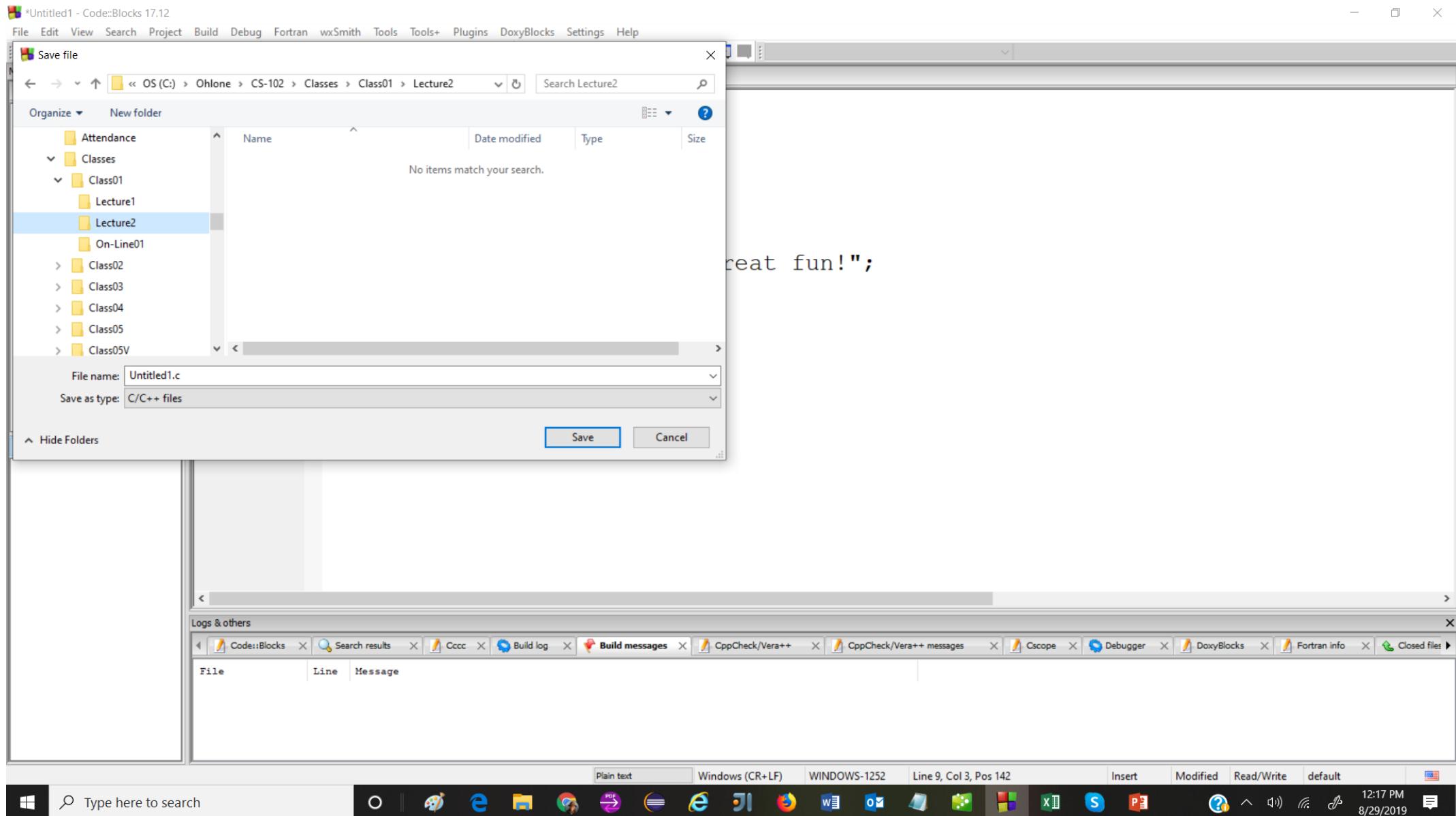
```
1 // A simple C++ program
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     cout << "Programming is great fun!";
8     return 0;
9 }
```

The code editor has a vertical yellow margin line on the left. The status bar at the bottom indicates the file is named "Untitled1", the encoding is "Plain text", the character set is "Windows (CR+LF)", the line is "Line 9, Col 3, Pos 142", and the file is "Modified". The taskbar at the bottom shows various application icons.

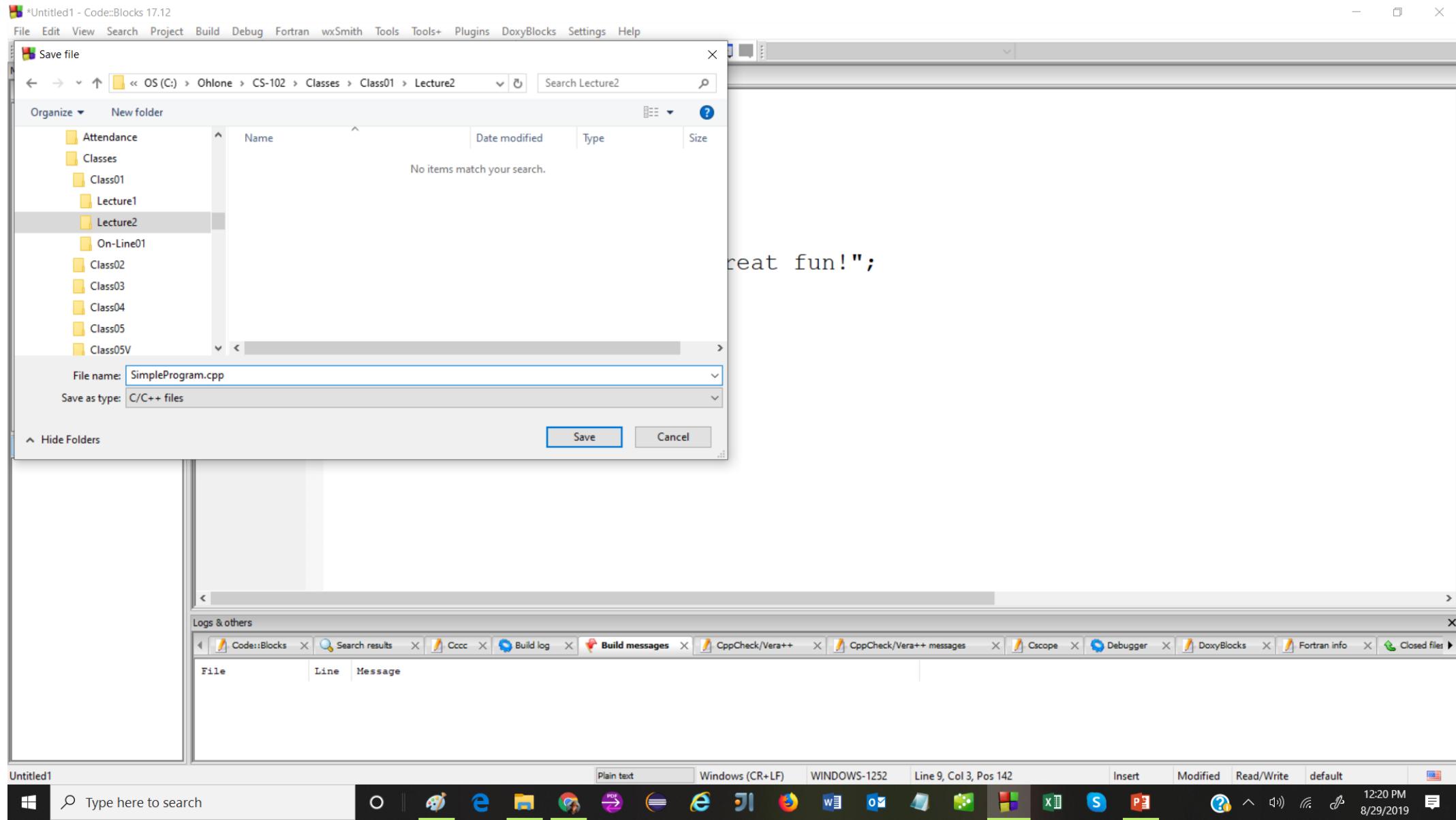
Do a File Save As



Don't Save it as a C - File.



Give it a Name and Save it as a .cpp file



Now it's a .cpp file and it is in Color

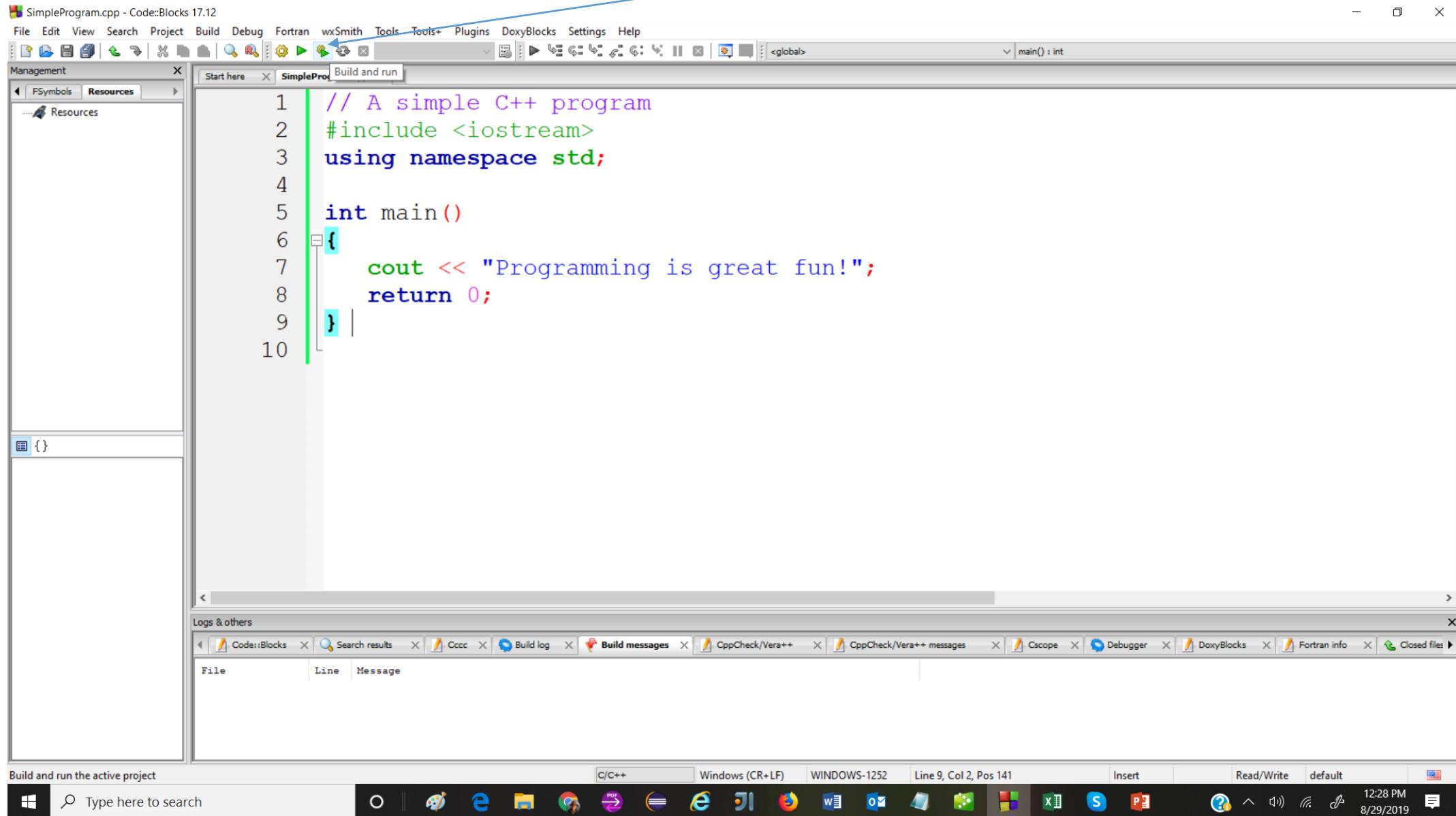
The screenshot shows the Code::Blocks 17.12 IDE interface. The main window displays a code editor with the file "SimpleProgram.cpp". The code is as follows:

```
// A simple C++ program
#include <iostream>
using namespace std;

int main()
{
    cout << "Programming is great fun!";
    return 0;
}
```

The code editor has syntax highlighting: comments are purple, keywords are blue, and strings are red. The project browser on the left shows a "Resources" folder. The bottom bar shows the file path "C:\Ohlone\CS-102\Classes\Class01\Lecture2\SimpleProgram.cpp", the current encoding "C/C++", and the status "Windows (CR+LF) WINDOWS-1252 Line 9, Col 2, Pos 141".

Now click on Build and Run



Compiled & Executed

The screenshot shows the Code::Blocks 17.12 IDE interface. The main window displays a code editor with the file "SimpleProgram.cpp" containing the following code:

```
1 // A simple C++ program
2 #include <iostream>
3 using namespace std;
```

Below the code editor is a terminal window showing the execution output:

```
C:\Ohlone\CS-102\Classes\Class01\Lecture2\SimpleProgram.exe
Programming is great fun!
Process returned 0 (0x0)   execution time : 13.821 s
Press any key to continue.
```

The terminal window has a title bar "C:\Ohlone\CS-102\Classes\Class01\Lecture2\SimpleProgram.exe". The status bar at the bottom right indicates the time as "12:31 PM" and the date as "8/29/2019".

Files Produced by Compiler

A screenshot of a Windows File Explorer window titled "Lecture2". The window shows a file structure on the left and a list of files on the right. The file structure includes "netfilter2", "Ohlone", "Correspondence", "CS-102", "Appendices", "Attendance", "Classes", "Class01", "Lecture1", "Lecture2" (which is selected), "On-Line01", "Class02", "Class03", "Class04", and "Class05". The list view on the right displays three files: "SimpleProgram.cpp" (C++ source file, 1 KB, modified 8/29/2019 12:23 PM), "SimpleProgram.exe" (Application, 1,528 KB, modified 8/29/2019 12:30 PM), and "SimpleProgram.o" (O File, 2 KB, modified 8/29/2019 12:30 PM). The status bar at the bottom shows "File Explorer" and "Windows File Explorer".

Name	Date modified	Type	Size
SimpleProgram.cpp	8/29/2019 12:23 PM	C++ source file	1 KB
SimpleProgram.exe	8/29/2019 12:30 PM	Application	1,528 KB
SimpleProgram.o	8/29/2019 12:30 PM	O File	2 KB

Changing the Default Extension Display: Select View/Options

Lecture2

File Home Share View

Panes

- Navigation pane
- Preview pane
- Details pane

Layout

- Extra large icons
- Large icons
- Medium icons
- Details

- Small icons
- List
- Tiles
- Content

Sort by

Group by

Add columns

Size all columns to fit

Item check boxes

File name extensions

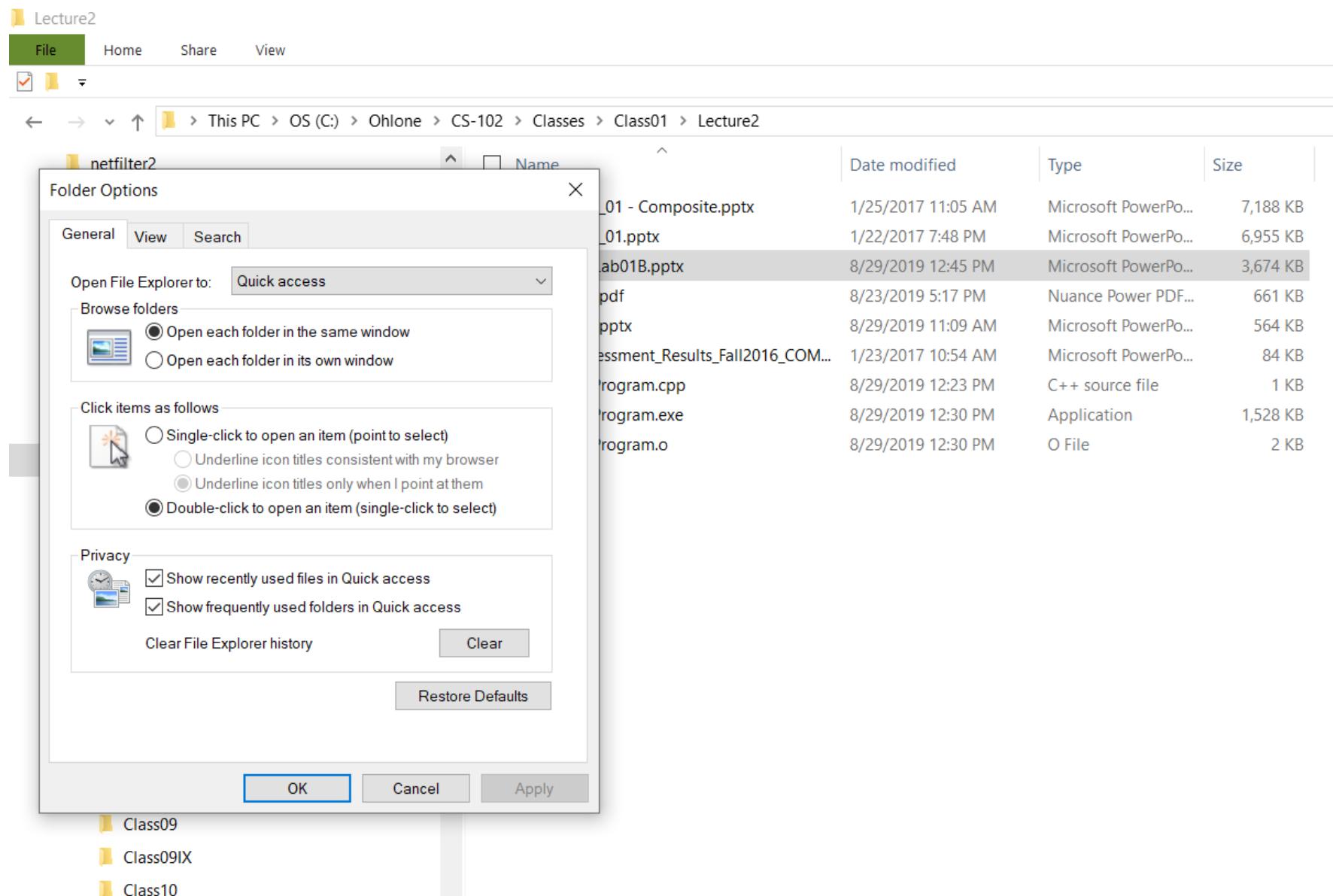
Hidden items

Hide selected items

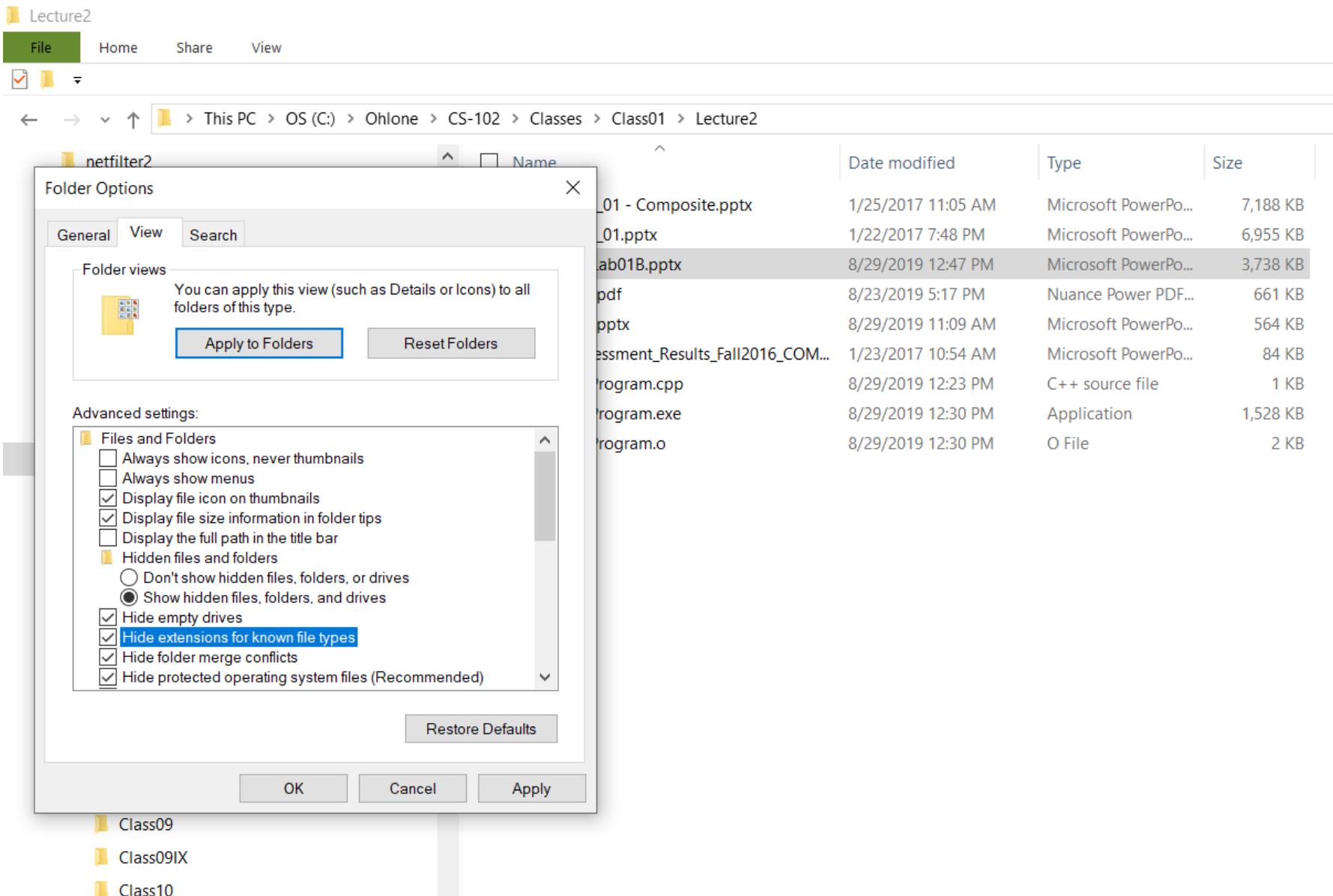
Options

File Name	Last Modified	Type	Size
Chapter_01 - Composite.pptx	1/25/2017	Change settings for opening items, file and folder views, and search.	7,188 KB
Chapter_01.pptx	1/22/2017		6,955 KB
IntroToLab01B.pptx	8/29/2019		3,613 KB
Lab01B.pdf	8/23/2019 5:17 PM	Nuance Power PDF...	661 KB
Lab01B.pptx	8/29/2019 11:09 AM	Microsoft PowerPo...	564 KB
Pre_Assessment_Results_Fall2016_COM...	1/23/2017 10:54 AM	Microsoft PowerPo...	84 KB
SimpleProgram.cpp	8/29/2019 12:23 PM	C++ source file	1 KB
SimpleProgram.exe	8/29/2019 12:30 PM	Application	1,528 KB
SimpleProgram.o	8/29/2019 12:30 PM	O File	2 KB

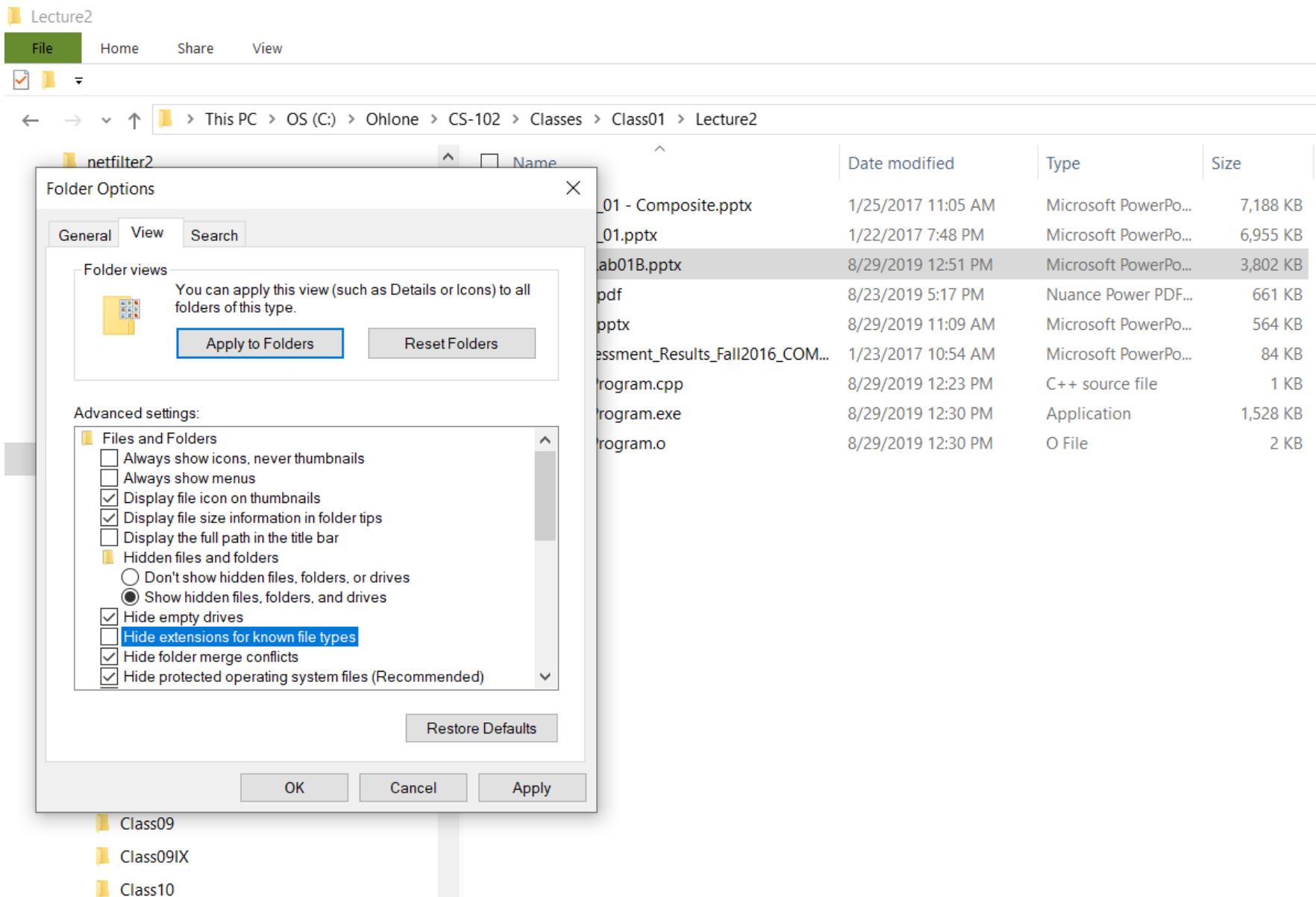
Now Select View



Note the Hide Extensions for known file types may be checked



Unhide the Extensions if it is clicked, Then click on OK



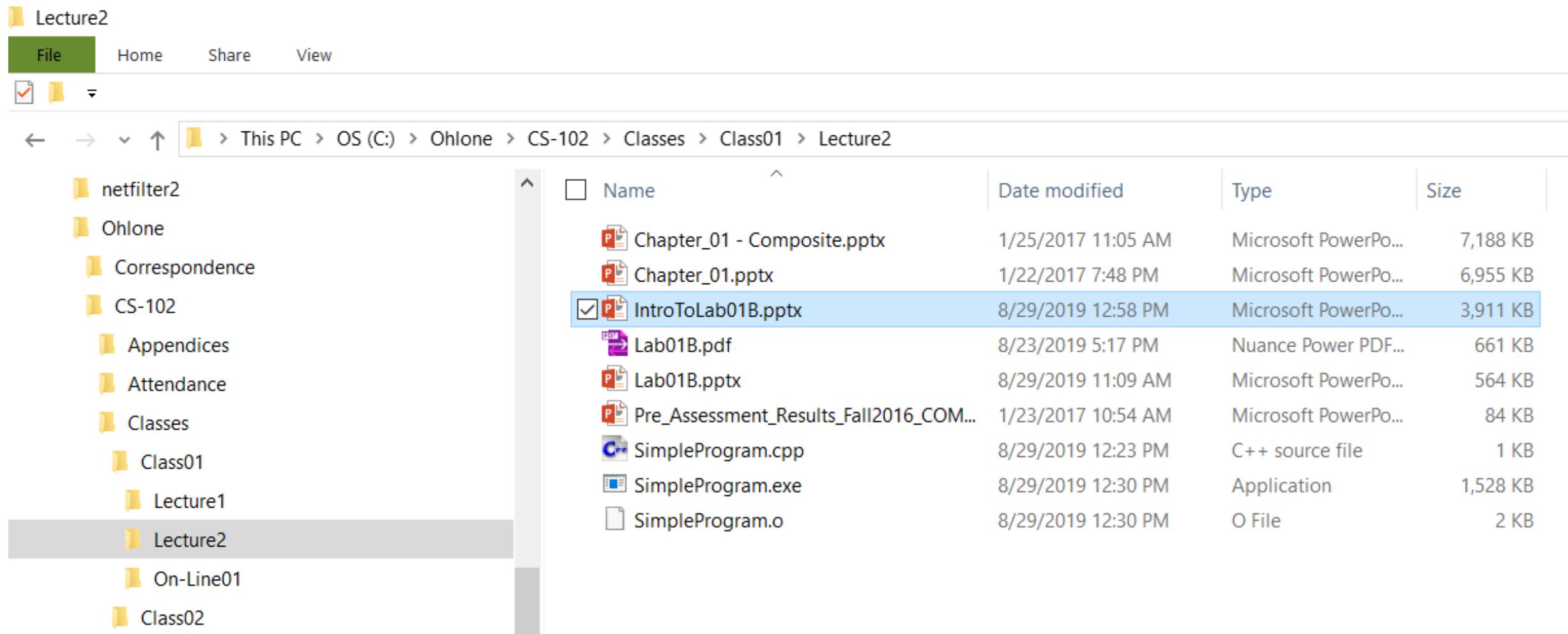
When Extensions are Hidden, here's what you see: Note that a file called “SimpleProgram” appears 3 times

The screenshot shows a Windows File Explorer window with the following details:

- Left pane (File Explorer tree):** Shows a hierarchy of folders: netfilter2, Ohlone, Correspondence, CS-102, Appendices, Attendance, Classes, Class01, Lecture1, **Lecture2** (selected), On-Line01, and Class02.
- Top menu bar:** File, Home, Share, View.
- Toolbar:** Back, Forward, Up, Refresh, Search, and a folder icon.
- Address bar:** This PC > OS (C:) > Ohlone > CS-102 > Classes > Class01 > Lecture2
- Right pane (List view):** A table showing files in the current folder. The columns are Name, Date modified, Type, and Size. The table includes the following rows:

Name	Date modified	Type	Size
Chapter_01 - Composite	1/25/2017 11:05 AM	Microsoft PowerPo...	7,188 KB
Chapter_01	1/22/2017 7:48 PM	Microsoft PowerPo...	6,955 KB
<input checked="" type="checkbox"/> IntroToLab01B	8/29/2019 12:55 PM	Microsoft PowerPo...	3,865 KB
Lab01B	8/23/2019 5:17 PM	Nuance Power PDF...	661 KB
Lab01B	8/29/2019 11:09 AM	Microsoft PowerPo...	564 KB
Pre_Assessment_Results_Fall2016_COM...	1/23/2017 10:54 AM	Microsoft PowerPo...	84 KB
SimpleProgram	8/29/2019 12:23 PM	C++ source file	1 KB
SimpleProgram	8/29/2019 12:30 PM	Application	1,528 KB
SimpleProgram.o	8/29/2019 12:30 PM	O File	2 KB

When they are Unhidden, here's what you see!
Now “SimpleProgram” appears as 3 distinct programs.



Let's open the Canvas Webpage for CS-102-04

ohlone.instructure.com/courses/17572

CS-102-04 Student View

2021FA Edit :

Home Course Status

Announcements Unpublish Published

Modules

Discussions

Welcome to: Intro to Computer Programming Using C++

CS-102

Import Existing Content

Import from Commons

Choose Home Page

View Course Stream

Course Setup Checklist

New Announcement

New Analytics

View Course Notifications

Coming Up View Calendar

Laboratory 1A

CS-102-04
10 points • Aug 31 at 4:10pm

OHLCONE COLLEGE

Account

Dashboard

Courses

Calendar

Inbox (111)

History

Commons

Studio

Help



Click on Modules on the Left

← → C ohlone.instructure.com/courses/17572 ⭐ 📄

 OHLONE COLLEGE

CS-102-04

2021FA

- Home
- Announcements
- Modules
- Discussions
- Grades
- Syllabus
- ConferZoom
- NetTutor
- Wellness Central
- BigBlueButton
(Formerly Conferences)
- Files
- Assignments
- Pages

63 Student View

21FA CS-102-04: Intro to Programming Using C++ (084359)

[Edit](#) [⋮](#)

Welcome to: **Intro to Computer Programming Using C++**

CS-102



Course Status

Unpublish Published

[Import Existing Content](#)

[Import from Commons](#)

[Choose Home Page](#)

[View Course Stream](#)

[Course Setup Checklist](#)

[New Announcement](#)

[New Analytics](#)

[View Course Notifications](#)

Coming Up [View Calendar](#)

 **Laboratory 1A**
CS-102-04
10 points • Aug 31 at 4:10pm

Clicking on Modules, here is what you'll see

← → ⌂ ohlone.instructure.com/courses/17572/modules ⌂

Announcements

Modules Start Here: Course Introduction ✓ + :

Discussions

Grades

Syllabus Course Syllabus ✓ + :

ConferZoom

NetTutor

Wellness Central Week 1 (Aug. 31 - Sept. 2) ✓ + :

BigBlueButton (Formerly Conferences) Week 2 (Sept. 7 - 9) ✓ + :

Files

Assignments

Pages Week 3 (Sept 14 - 16) ✓ + :

Quizzes

Rubrics

People Week 4 (Sept. 21 - 23) ✓ + :

Outcomes

Collaborations

OHLCONE COLLEGE

Account

Dashboard

Courses

Calendar

Inbox

History

Commons

Studio

Help

Clicking on the Down Arrow for Week1, we see:

The screenshot shows a user interface for a learning management system. On the left, there is a vertical navigation bar with various icons and links. The 'Courses' link is highlighted in orange. Below it, the 'Dashboard' link is also highlighted in orange. Other visible links include 'Account', 'Syllabus', 'ConferZoom', 'NetTutor', 'Wellness Central', 'BigBlueButton (Formerly Conferences)', 'Files', 'Assignments', 'Pages', 'Quizzes', 'Rubrics', 'People', 'Outcomes', 'Collaborations', 'New Analytics', and 'Item Banks'. A yellow box highlights the 'Inbox' icon, which has a red badge with the number '111'. At the bottom left is a back arrow icon.

The main content area displays the course syllabus and the first week's assignments. The syllabus section includes a 'Course Syllabus' item. The 'Week 1 (Aug. 31 - Sept. 2)' section is expanded, showing the following items:

- Week 1 Overview
- Chapter_01_Slides.pdf
- Quiz 0 (Aug 31 | 6 pts)
- Laboratory 1A (Aug 31 | 10 pts)
- Laboratory 1B (Sep 2 | 10 pts)
- Homework 01 (Sep 7 | 20 pts)
- Closing Remarks For Week 1

Each assignment item has a green checkmark icon, a plus sign icon, and a three-dot menu icon.

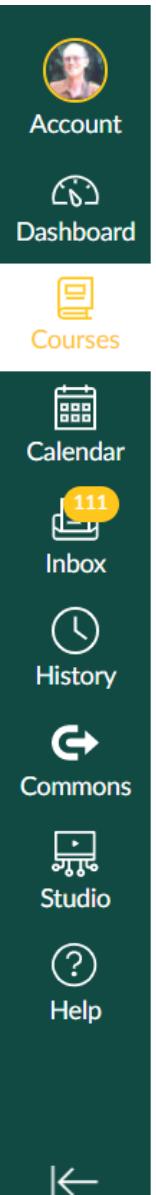
Scrolling down to the bottom of Modules, we see:

The screenshot shows a Moodle course page titled "ohlone.instructure.com/courses/17572/modules". The left sidebar contains navigation links: OHLONE COLLEGE, Account (with a profile picture), Dashboard, Courses (highlighted in orange), Calendar, Inbox (with 111 notifications), History, Commons, Studio, and Help. The main content area displays a list of course modules:

- Week 12 (Nov. 16 - 18)
- Week 13 (Nov. 23)
- Week 14 (Nov. 30 - Dec. 2)
- Week 15 (Dec. 7 - 9)
- Final Exam Week (Dec. 16)
- Appendices

Each module item has a green checkmark icon, a plus sign, and a three-dot menu icon.

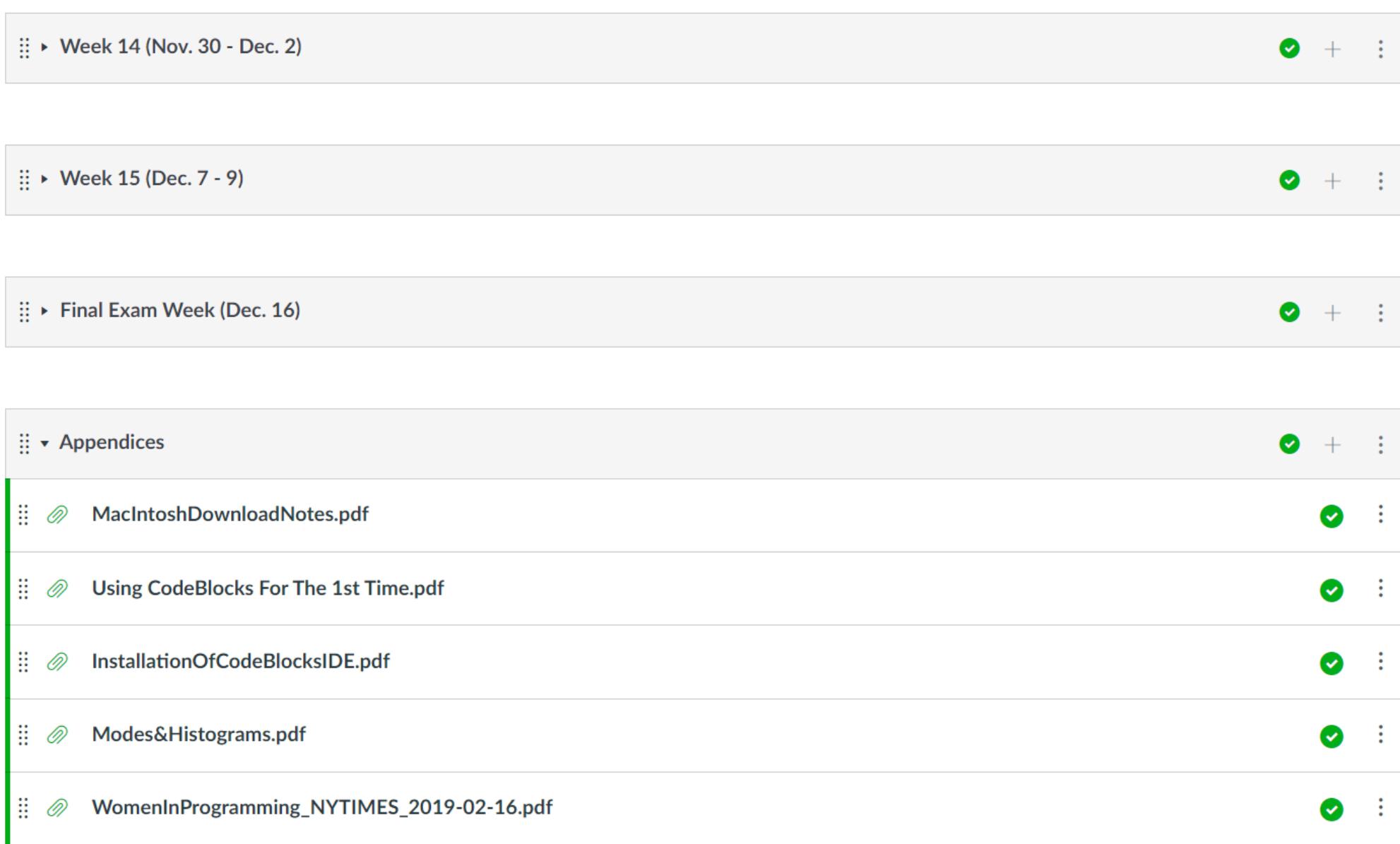
Here we will see how to download a C++ IDE



A vertical sidebar menu with icons and labels:

- Account (User icon)
- Dashboard (Clock icon)
- Courses (Book icon)
- Calendar (Calendar icon)
- Inbox (Envelope icon with 111 messages)
- History (Clock icon)
- Commons (Circular arrow icon)
- Studio (Monitor icon)
- Help (Question mark icon)

At the bottom is a large left-pointing arrow icon.



The main area shows course navigation and a file list:

- Week 14 (Nov. 30 - Dec. 2)
- Week 15 (Dec. 7 - 9)
- Final Exam Week (Dec. 16)

Under Appendices:

- MacintoshDownloadNotes.pdf
- Using CodeBlocks For The 1st Time.pdf
- InstallationOfCodeBlocksIDE.pdf
- Modes&Histograms.pdf
- WomenInProgramming_NYTImES_2019-02-16.pdf