

# C++ Review

## Lecture 1

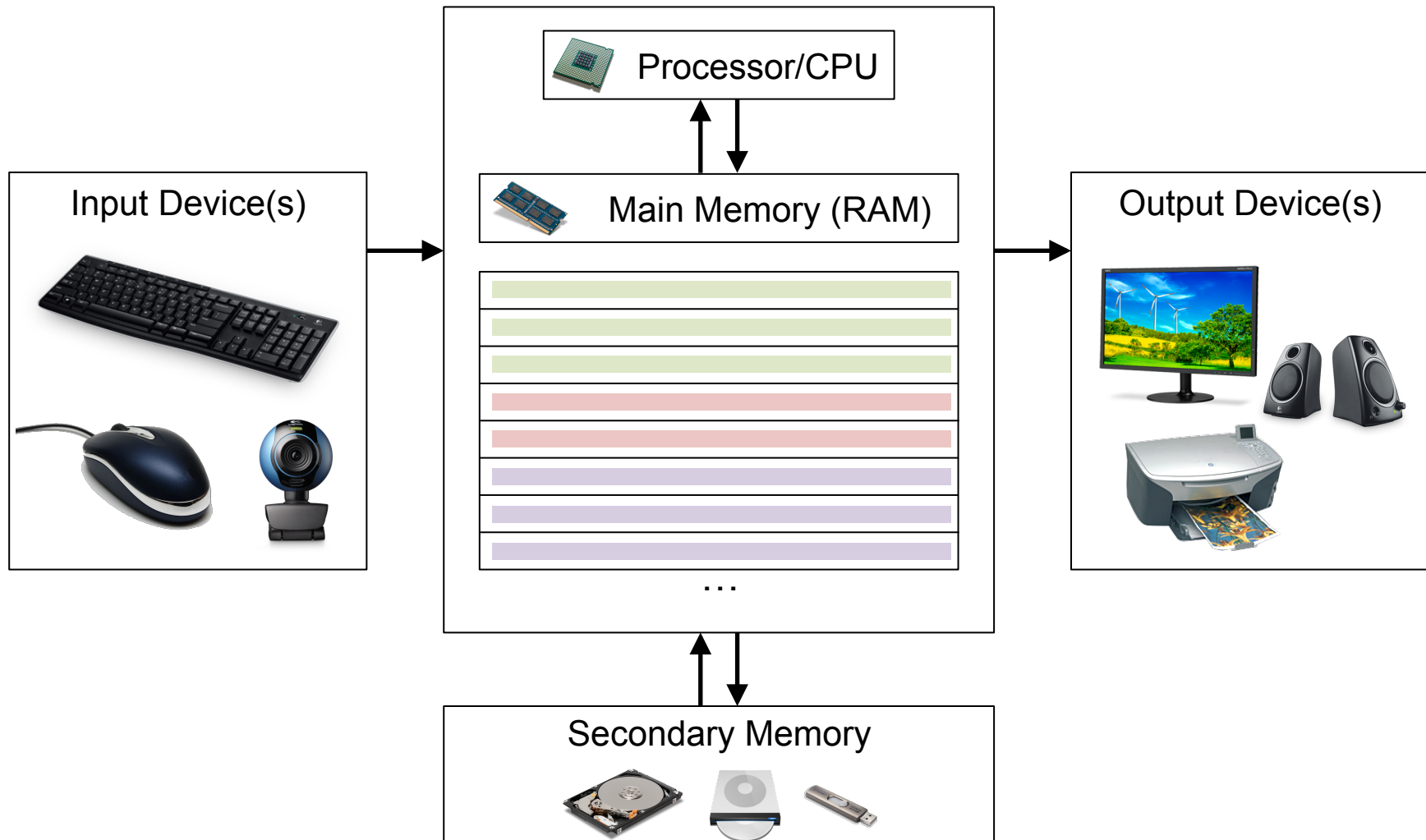


# Your Responsibility

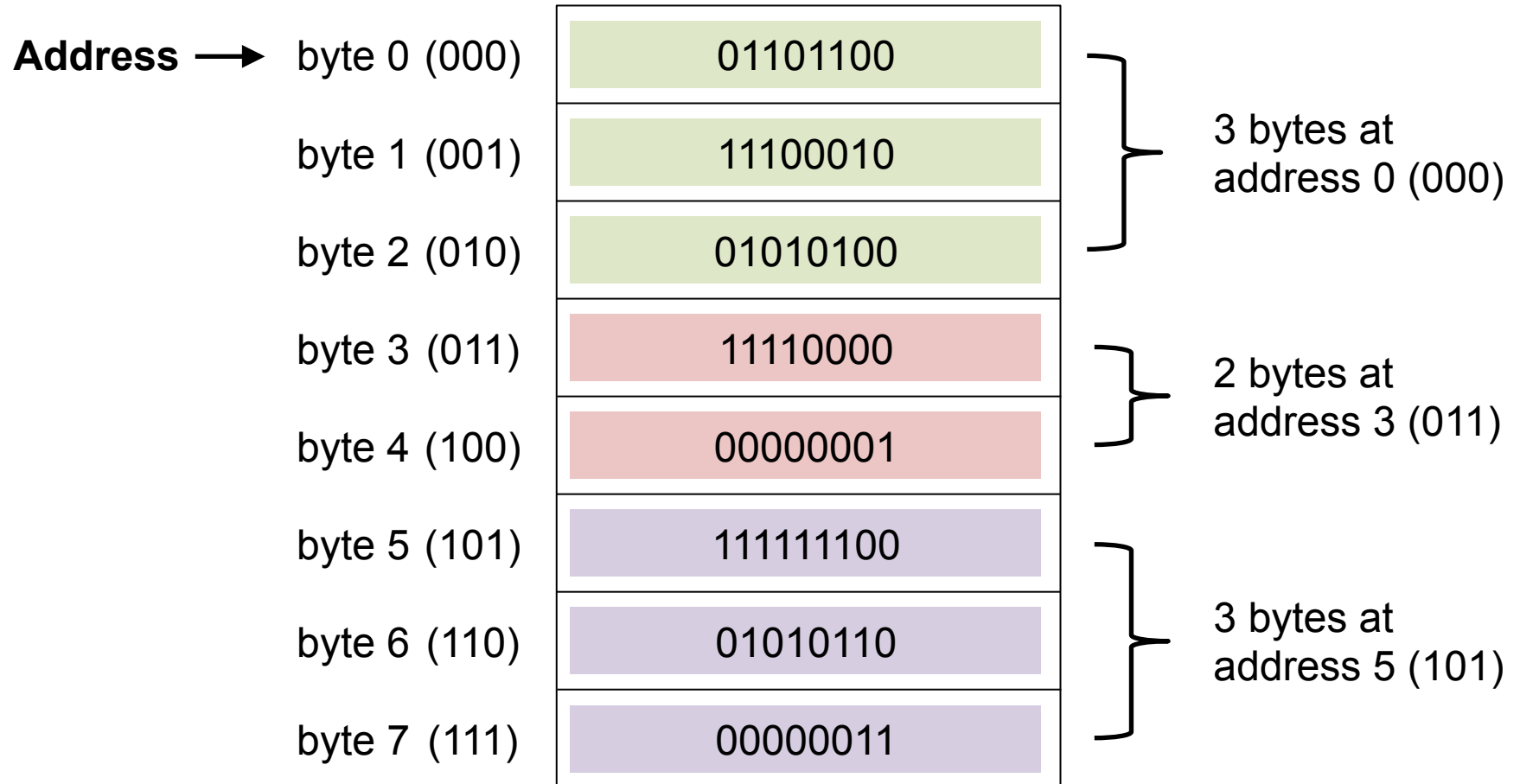
- This review is **NOT** exhaustive
  - Just brushing away some cobwebs from the break
- All COMP128 material is fair game for any assignment/exam
  - See BB for slides
  - Review chapters in the book



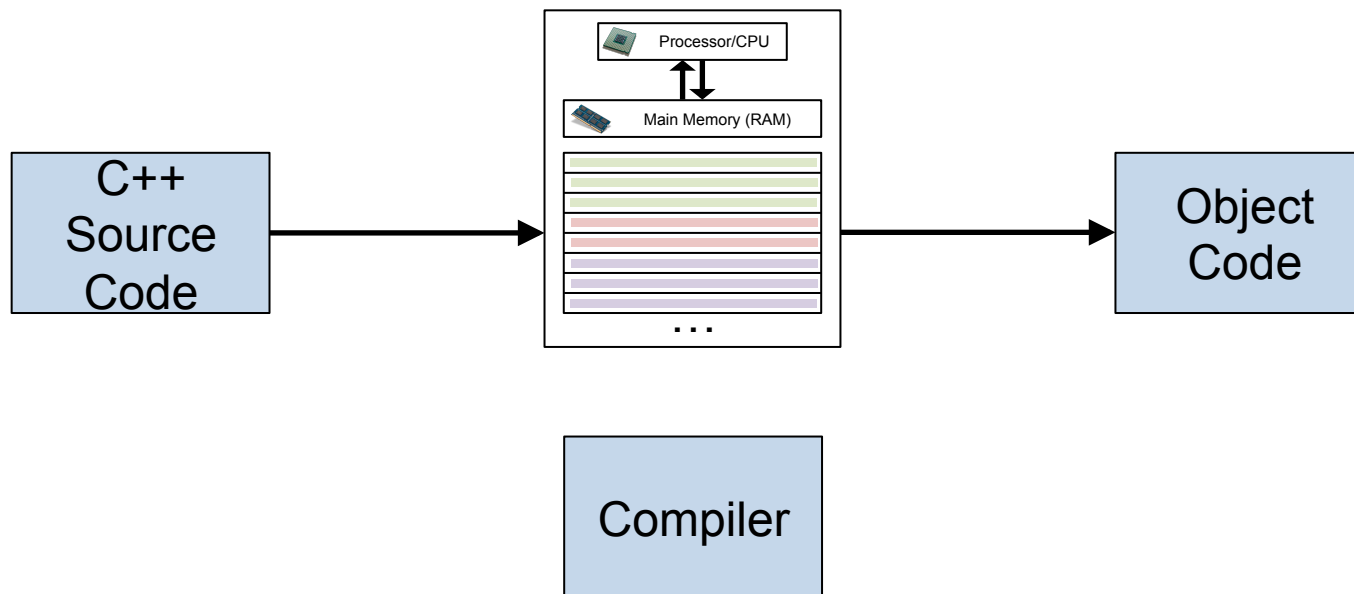
# High-Level View of Hardware



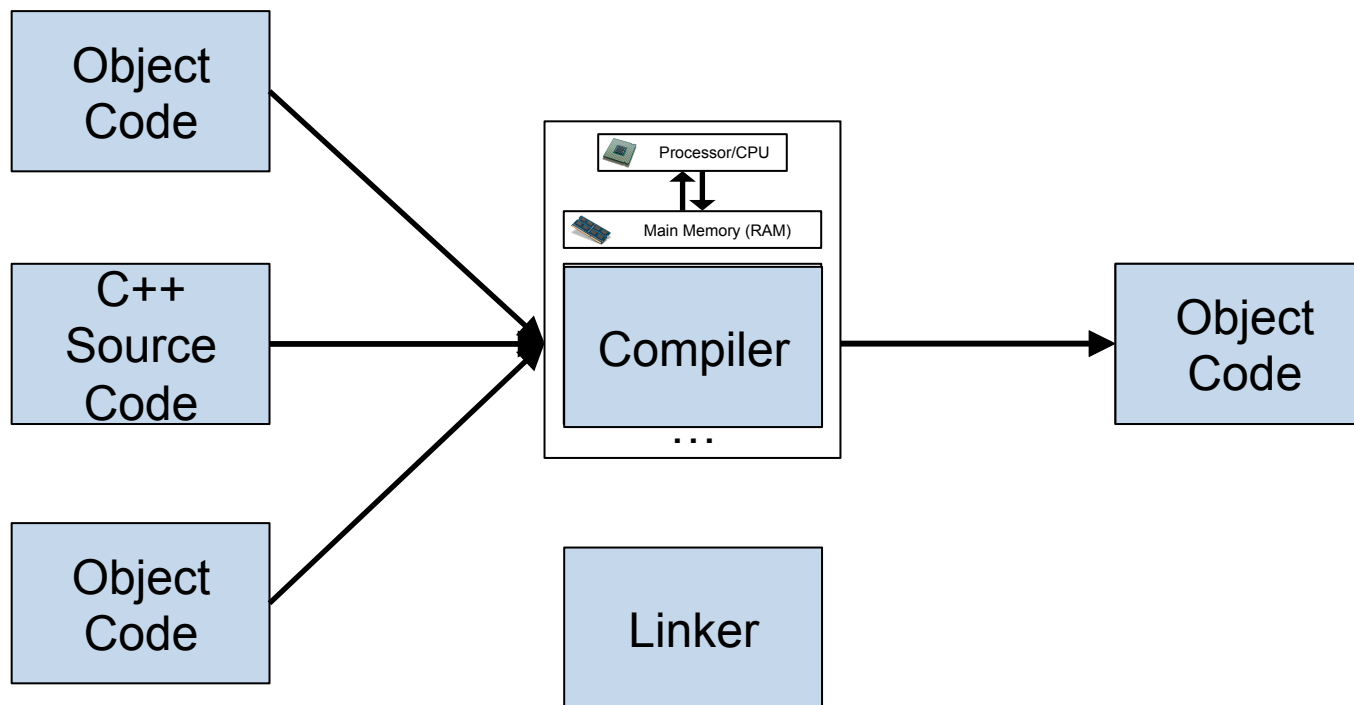
# Main Memory (RAM)



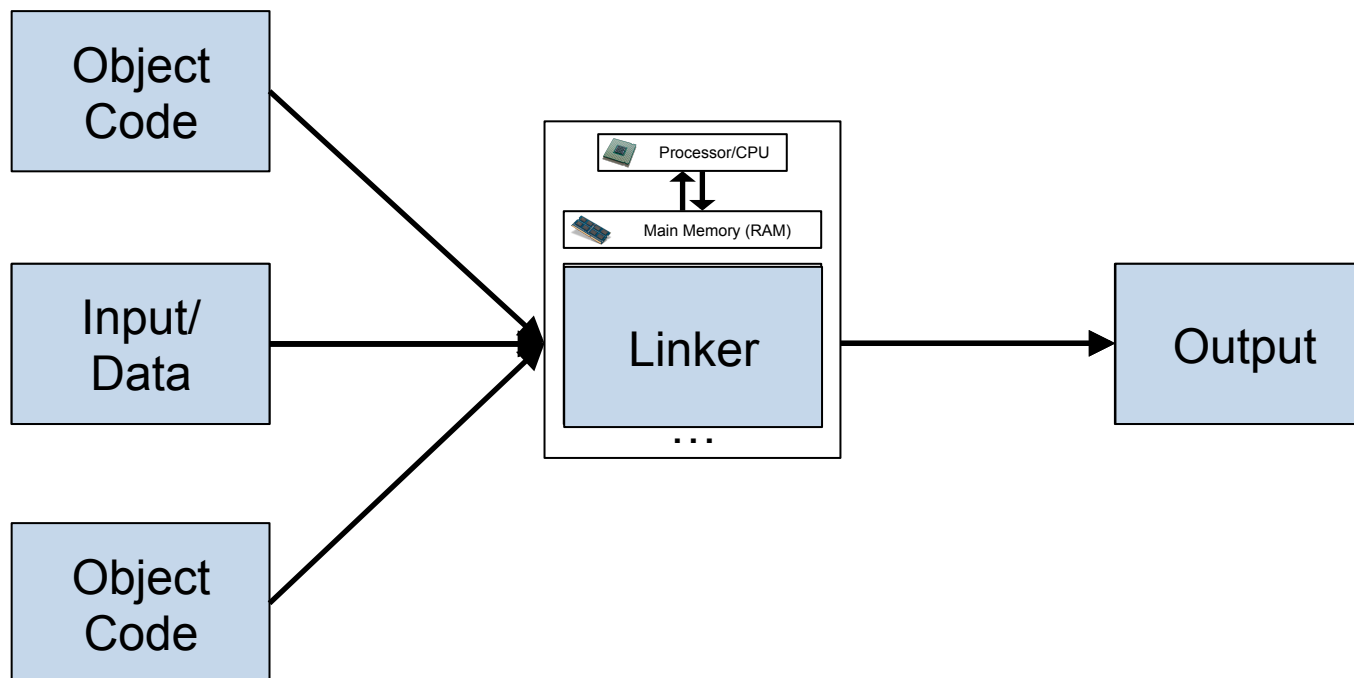
# Build and Run (1)



# Build and Run (2)



# Build and Run (3)



# Build Errors

## Compiler

- Typically **syntax** errors
  - Forgot a semi-colon
  - Mistyped a command
  - Forgot an argument
  - Incorrect type
- Forgot `#include`
- Couldn't find library (more later)

## Linker

- Undefined symbols
  - Declared a function, forgot to define (or changed args)
  - Forgot namespace
- Cannot write executable (usually still running)
- Couldn't find library (more later)





# Output

Write a program that prints “Hello World” to the screen (sans quotes)



# Answer

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

int main()
{
    cout << "hello world" << endl;
    return 0;
}
```



# I/O, Variables, Expressions

Write a program that asks the user for an integer and then prints out that number to the screen, a tab, and then that number plus five. For example:

```
Please enter a number: 5
5 10
```



# Answer

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

int main()
{
    int x;
    cout << "Please enter a number: ";
    cin >> x;
    cout << x << "\t" << ( x+5 ) << endl;
    return 0;
}
```




# Flow Control

What is wrong with the following code, why?

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

int main()
{
    int x, y;
    y = x + 5;
    cout << "Please enter a number: ";
    cin >> x;
    cout << x << "\t" << y << endl;
    return 0;
}
```



# If/Else

Write a program that asks the user for a numeric grade (could be a fraction) and outputs the Wentworth letter grade (no rounding).

Grade	Definition	Weight	Numerical
A	Student learning and accomplishment far exceeds published objectives for the course/test/assignment and student work is distinguished consistently by its high level of competency and/or innovation.	4.00	96 - 100
A-		3.67	92 - 95
B+	Student learning and accomplishment goes beyond what is expected in the published objectives for the course/test/assignment and student work is frequently characterized by its special depth of understanding, development, and/or innovative experimentation.	3.33	88 - 91
B		3.00	84 - 87
B-	Student learning and accomplishment meets all published objectives for the course/test/assignment and the student work demonstrates the expected level of understanding, and application of concepts introduced.	2.67	80 - 83
C+		2.33	76 - 79
C		2.00	72 - 75
C-	Student learning and accomplishment based on the published objectives for the course/test/assignment were met with minimum passing achievement.	1.67	68 - 71
D+		1.33	64 - 67
D		1.00	60 - 63
F	Student learning and accomplishment based on the published objectives for the course/test/assignment were not sufficiently addressed nor met.	0.00	< 60



# Answer

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

int main()
{
    double grade;
    cout << "Please enter a grade: ";
    cin >> grade;

    if ( grade >= 96 )
    {
        cout << "A" << endl;
    }
    else if ( grade >= 92 )
    {
        cout << "A-" << endl;
    }
    else if ( grade >= 88 )
    {
        cout << "B+" << endl;
    }
    else if ( grade >= 84 )
    {
        cout << "B" << endl;
    }
    else if ( grade >= 80 )
    {
        cout << "B-" << endl;
    }
    else if ( grade >= 76 )
    {
        cout << "C+" << endl;
    }
    else if ( grade >= 72 )
    {
        cout << "C" << endl;
    }
    else if ( grade >= 68 )
    {
        cout << "C-" << endl;
    }
    else if ( grade >= 64 )
    {
        cout << "D+" << endl;
    }
    else if ( grade >= 60 )
    {
        cout << "D" << endl;
    }
    else
    {
        cout << "F" << endl;
    }
    return 0;
}
```



# Functions, Loops

Write a function named `nums` that takes two integer arguments: `upper_bound` and `per_line` and prints to the screen each number from 1 to `upper_bound`, `per_line` numbers on a line.

```
nums( 12, 5 );
```

```
1 2 3 4 5
```

```
6 7 8 9 10
```

```
11 12
```





# Answer

```
void nums(int upper_bound, int per_line)
{
    int num_on_line = 0;
    for ( int i=1; i<=upper_bound; i++ )
    {
        cout << i << " ";
        num_on_line++;
        if ( num_on_line >= per_line )
        {
            num_on_line = 0;
            cout << endl;
        }
    }
}
```



# Functions, Arrays, Pass by Reference

Write a function `min_max_avg` that takes as arguments an array of double variables (`arr`) and two double variables, by reference (`minval`, `maxval`). The function sets these variables to be the smallest and largest values in the array and returns the average of all the numbers.



# Answer

```
double min_max_avg(double arr[], int count, double& minval, double& maxval)
{
    minval = arr[0];
    maxval = arr[0];
    double sum = arr[0];

    for ( int i=1; i<count; i++ )
    {
        if ( arr[i] < minval )
            minval = arr[i];

        if ( arr[i] > maxval )
            maxval = arr[i];

        sum += arr[i];
    }

    return ( sum / count );
}
```



# Files, Loops

Write a program that reads a file of characters named “chars.txt” and, whenever it encounters a letter, outputs the uppercase version of that letter, one per line, and ignores all other characters.



# Answer

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

int main()
{
    ifstream f;
    f.open( "chars.txt" );
    if ( f.fail() )
    {
        cout << "Error opening chars.txt" << endl;
        return 1;
    }

    char c;
    while ( f >> c )
    {
        if ( c >= 'a' && c <= 'z' )
            c -= ( 'a' - 'A' );

        if ( c >= 'A' && c <= 'Z' )
            cout << c << endl;
    }

    f.close();
    return 0;
}
```



# Wrap Up

- These exercises should have been easy; if you had difficulties, please look back to your COMP128 materials, the book, and/or come to office hours
- This review didn't cover classes; there will be dedicated lectures reviewing and expanding on them in a couple weeks (continuing for much of the semester!)
- All of this material (and more) is fair game for any assignment/exam

