Functions, Scope & File IO

C++ Lecture 4

Nick Matthews

Adam Kohl

Iowa State University
VRAC Visualize • Reason • Analyze • Collaborate

What we hope to learn today

- Putting everything into a main file isn't feasible
- Blocks of code that is easily referenced and usable
- What happens with repeated variable names?
- File I/O – Simple how do we read and write to a file?*

* If we get to it

Functions

- A **function** is a group of statements that together perform a task.
- You have already come across at least one.

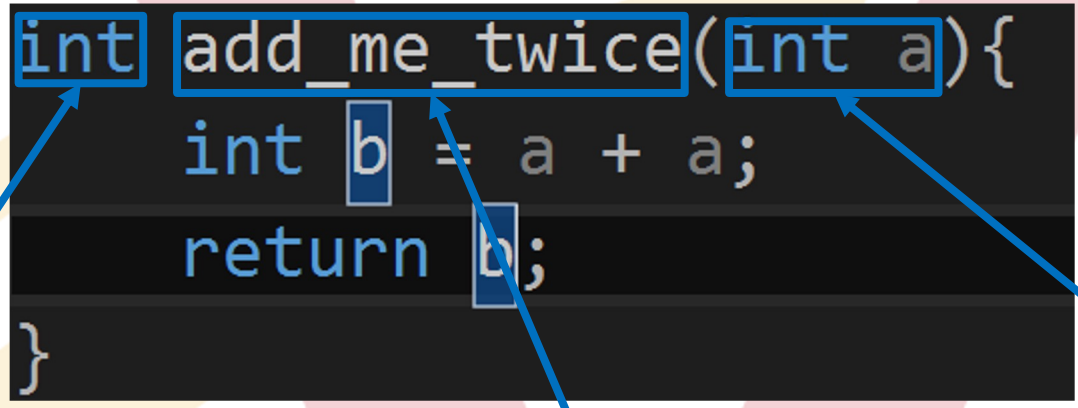
```
int main(){
    //Execute something
    return 0;
}
```

Functions

- Helps to separate out code instead of putting everything in one place
- Makes things more organized and easily referenceable
- Scope changes* help in keeping less cluttered memory

*We will learn what scope is later

Function definition



Return value
Returns only one thing at a time
Can be anything (int, char, double)

Function name has to be a new name never defined before with the same parameters

Input parameters

When should we use functions?

- If you repeat code always use functions
- Always use function names that represent what the function does (good coding practice)
  - Int division_by_2 (int num) { return num / 2; } //Good code
  - Int abcd(int num) { return num /2 ; } //Terrible code
- Naming doesn't pertain only to functions though – variables as well

Mini Task

Open Functions.cpp

Make a new function add_me_three_times

Make a new function add_me_four_times that uses add_me_twice

Passing multiple variables

- We've seen only one variable passed to a function
- Passing multiple variables is pretty much the same way
  - int func(int a, int b) { … }
- You can pass as many as you want
- Good coding practice is to use not more than 3 inputs

Functions and arrays

- Passing an array to a function
  - void func (int array[]) { … }
- There are other ways also
  - void func (int* array) { … } //using pointers
- For now we will use the first way only

Function defaults

- Sometimes you want the ability for functions to have default values
  - Void func (int a, int b = 2) { a = 3; b = b * 2;}
- Defaults have to be declared at the end of the function
- Check out what happens if you do it in the beginning

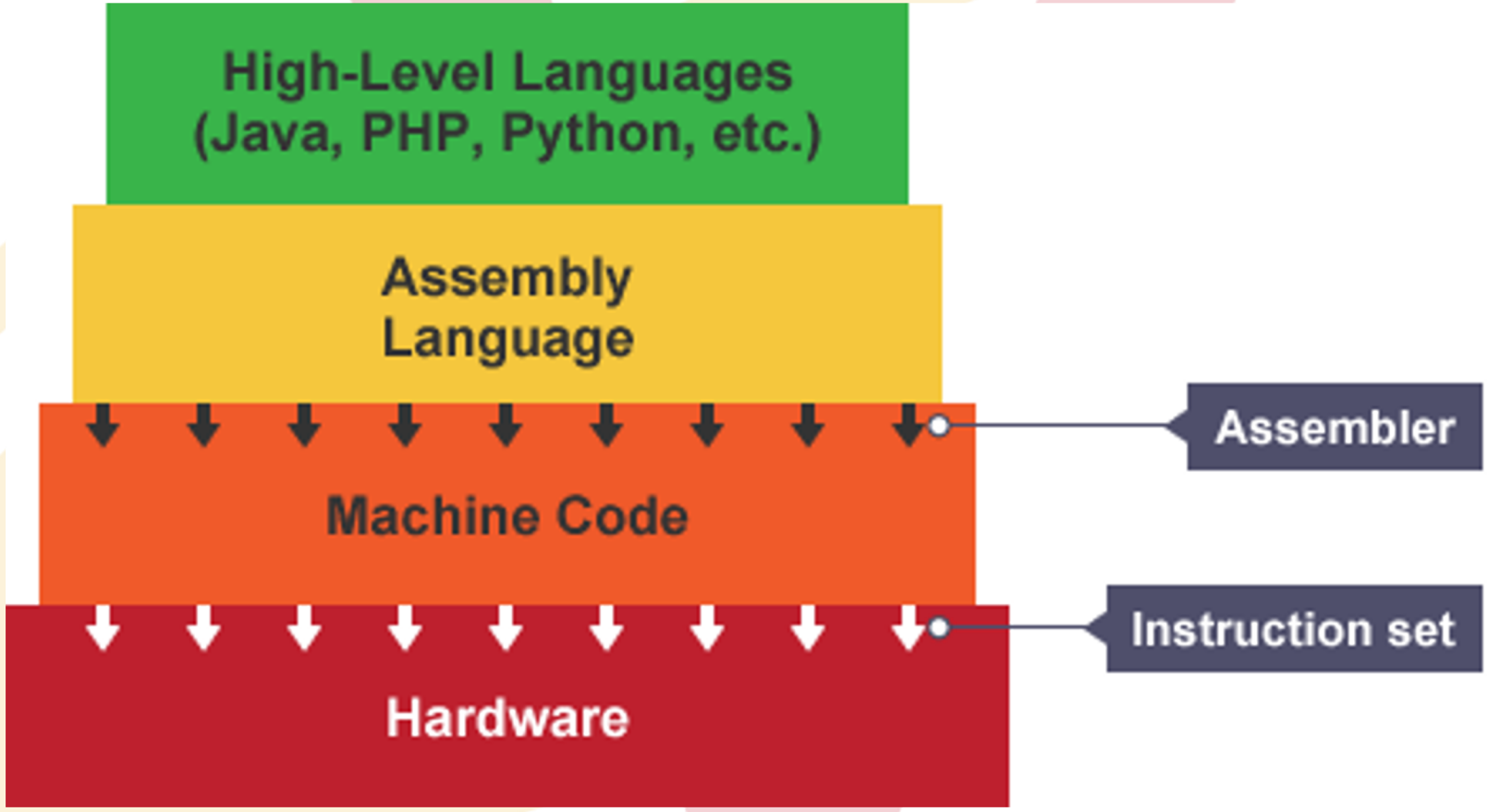
Mini Task

Write a function that finds the average of a size 100 array of integers (int) unless otherwise specified

Scope

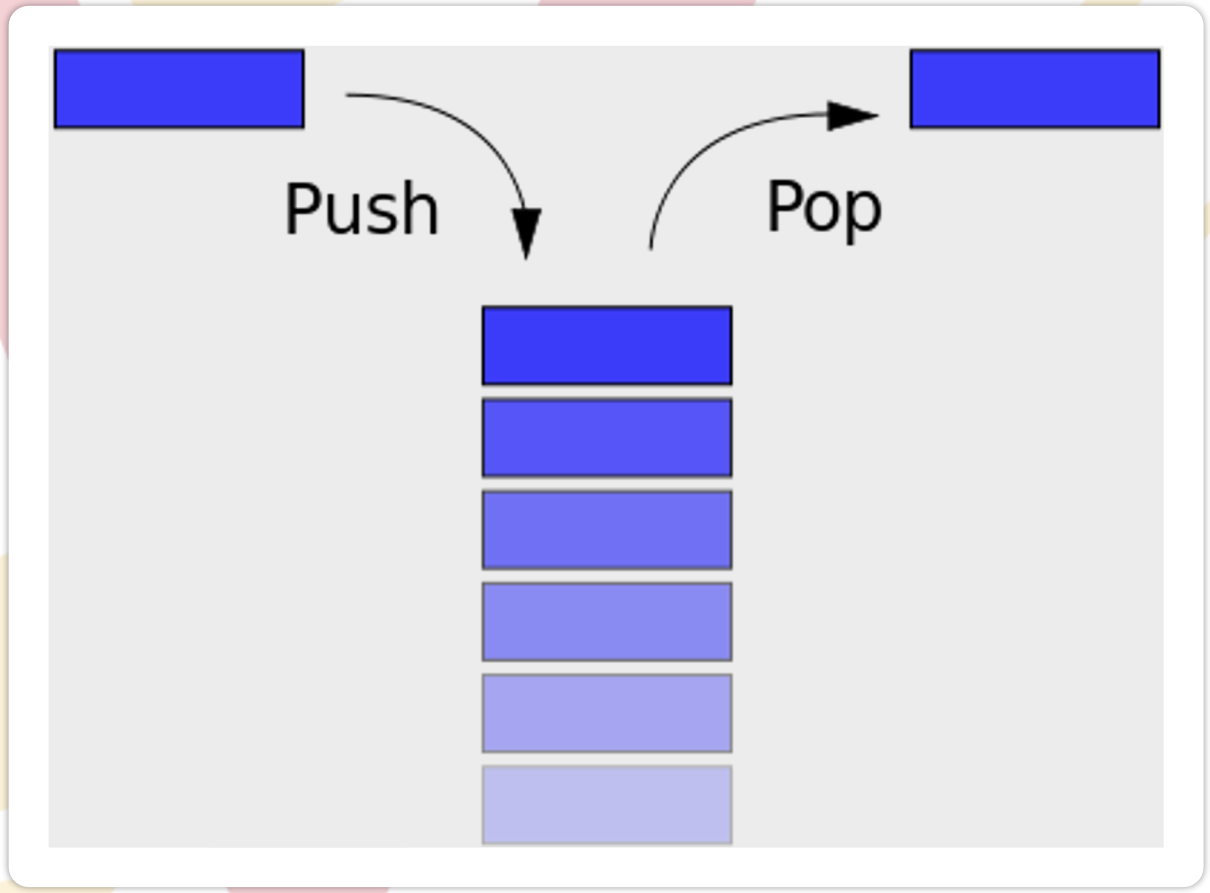
- Slightly complicated concept
- Born out of memory necessity
- Now useful to keep things easily organized

How does any program run?



Program stack

- Part of the memory where all things static are stored as and used when needed
- Stack is LIFO



How does the code run?

```
int x = 5;
double y = 10;
for (int i = 0; i <= 3; i++){
    int y = x;
    x = x + 1;
    cout << x << " " << y << endl;
}
```

```
int x = 5;
double y = 10;
for (int i = 0; i <= 3; i++){
    int y = x;
    x = x + 1;
    cout << x << " " << y << endl;
}
```

program

Y = 5
I = 0
Y = 5
X = 6

Mini Task

We will walk through it together

Break points and value checking

FILE I/O

- Add ways to read and write to files
- Why do we want to do that?
- Standard libraries help us out and do a lot of the heavy lifting

```
fstream fpout("test.txt",ios::out);
fpout << "New Text" << endl;
fpout << "Newer Text" << endl;
fpout.close();
```

Variable name of the file handler

File name

Type of access Mode Flag

Same way that we use cout can be used here

FILE I/O – Reading from a file

- Pretty much the same concept
- Only some things are flipped

```
std::string str="";
fstream fpin;
fpin.open("test.txt",ios::in);
fpin >> str;
cout << str << endl;
fpin >> str;
cout << str << endl;
fpin.close();
```

Mode Flag	Description
ios::app	Append mode. All output to that file to be appended to the end.
ios::ate	Open a file for output and move the read/write control to the end of the file.
ios::in	Open a file for reading.
ios::out	Open a file for writing.
ios::trunc	If the file already exists, its contents will be truncated before opening the file.

Mini Task

Execute the file part of Functions.cpp

Does everything work?

Questions?

Assignment

- Do the functions assignment.cpp
- If you finish quickly enough do the advanced one
- If you finish that let me know and I will give you some more challenging stuff