C++ Workshop 1
Basic Code Structures

Panagiotis Petridis

Brought to you by HackSoc 

Variable Declarations

var_type var_name = value;

Available types:

- int
- unsigned
- short
- float
- char
- double
- long [long]
- string*

```
int a = 0;
int b = 3;
a += 5;
b = b - a * 3;
```

Arrays

type var_name[] = {value, value,
…, 22, 1};

type var_name[3] = {value, value, value};

type var_name[constant value];

Indexing

C++ does NOT do *bound checking*!!!

type var_name[3] = {value, value, value};

var_name[4] = ??
Segmentation Fault

If-else statements

```
if (condition) {
    // code…
} else if (another_condition) {
    // code…
} else {
    // code…
}
```

```
if (true) {
    // code…
} else if (true || (1 && false) ){
    // code…
} else {
    // code…
}
```

Loops

```
for (statement; condition; statement) {
    // code…
}

while (condition) {
    // code…
}

do {
    // code…
} while (condition);
```

```
for(int i = 0; i < 10; i ++) {

}

while (true) {

}

do {
    // code…
} while (false);
```

Switch statements

```
switch(variable) {
    case value:
        // code…
        break;
    case value:
        // code…
        break;
    default:
        // code…
        break;
}
```

```
switch(error_code) {
    case 200:
        // OK
        break;
    case 404:
        // NOT FOUND
        break;
    default:
        // PANIC
        break;
}
```

Functions

```
type func_name(type p1, type p2, …) {
    return value;
}
```

```
void do_stuff (int a, string s) {
    return a + s.size();
}
```

```
int main() {
    // code…
    return 0;
}
```

```
void  exit_early() {
    // code will run
    return;
    // code won’t run
}
```

Structs

```
struct name {
    type var;

    name()  {
        // initialize
    }

    type fun() {
        return value;
    }
};
```

```
struct name {
    type first_name;
    type last_name;

    type full_name() {
        return first_name + last_name;
    }
};
```

Classes (fancy structs)

```
class Animal {
public:
    Animal(type var, ...) { ... }
    type fun() { ... }
    type var; // don't do this
private:
    type fun_fun() { ... };
    type var;
    type var;
};
```

```
class Animal { ... };



int main() {
    Animal doggo("Ector", "woof");
    doggo.make_sound();
    return 0;
}
```

#include libraries

You can include libraries that exist in your PATH with:

```
#include <library_name>
```

```
#include <iostream>
#include <vector>
```

To include code from other files you can use:

```
#include "library_name.cpp"
#include "library_name.h"
```

```
#include "my_algorithms.cpp"
#include "my_cool_class.h"
```

Namespaces

```
using namespace my_namespace;

namespace my_namespace {
   // separated code…
}
```

```
using namespace std;
namespace my_namespace {
   void my_function() { … }
}

my_namespace::my_function();
```

Output to console

```
#include <iostream>

using namespace std;

int main() {
    cout << "Hello World!" << endl;
    return 0;
}
```

I/O stream

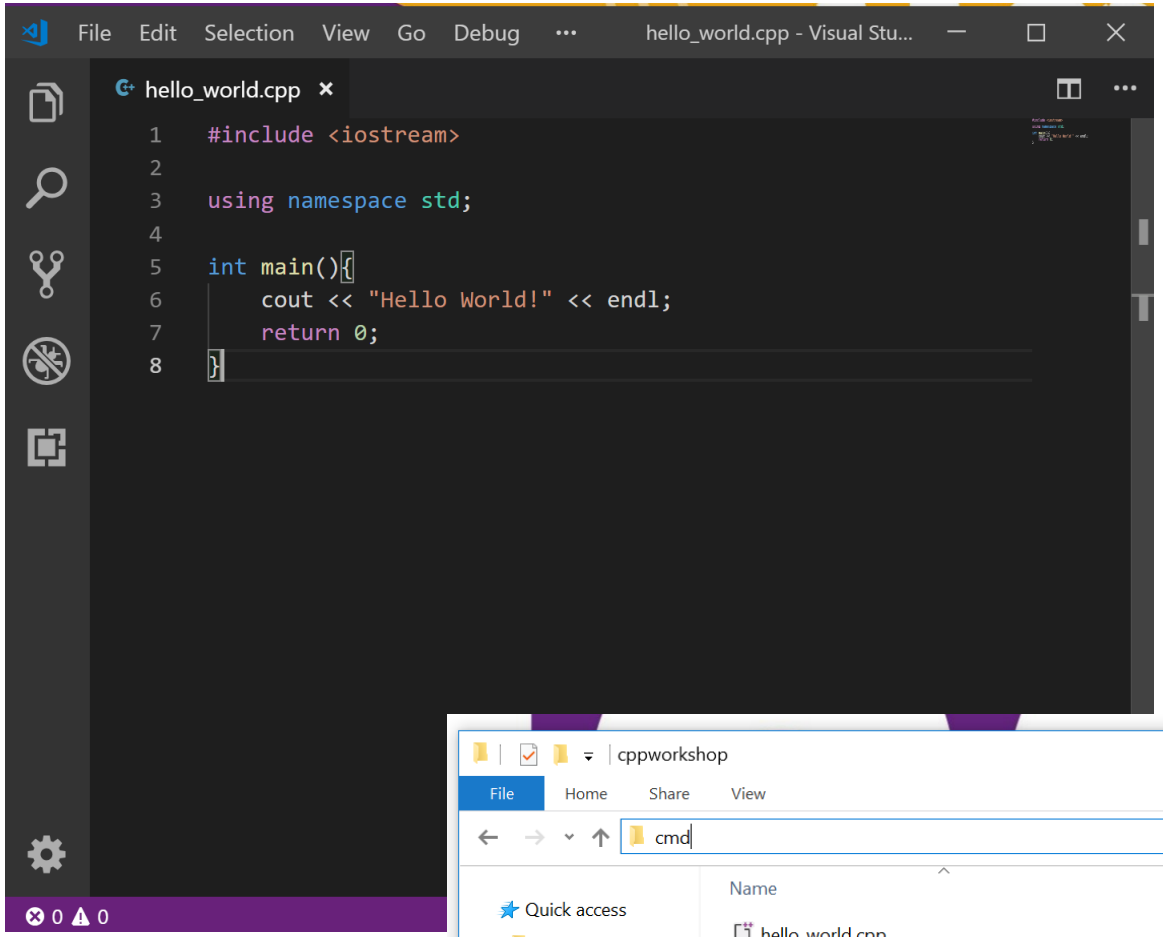
```
#include <iostream>

int main() {
    int a = 7;
    std::cout << "Lucky number: " << a
    << std::endl;
    return 0;
}
```

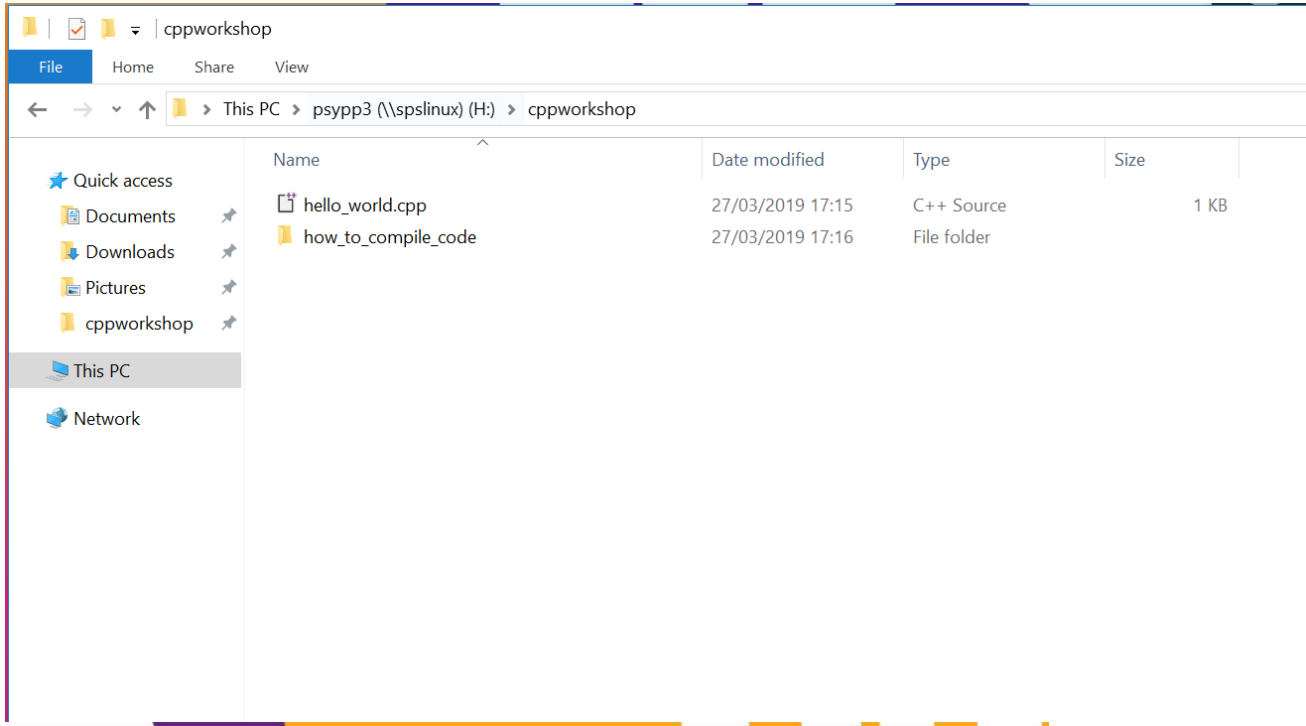
User input

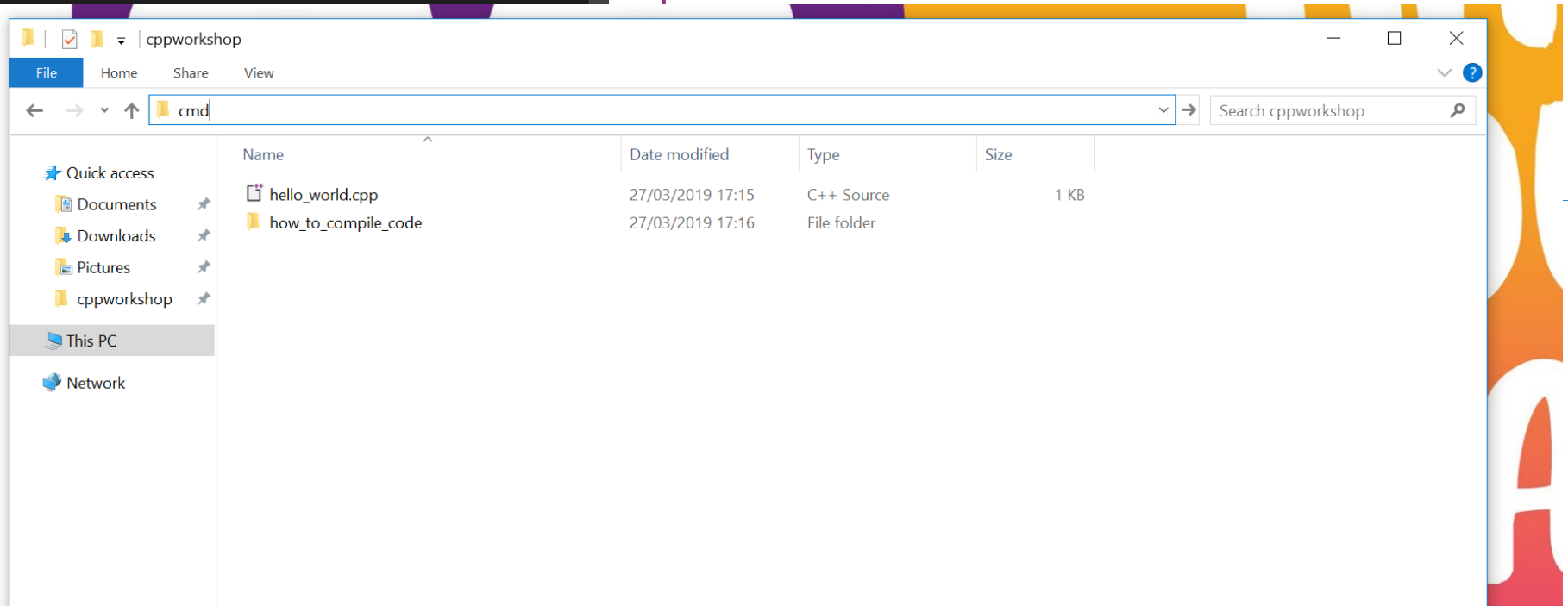
```
#include <iostream>

using namespace std;

int main() {
    string name;
    cin >> name;
    cout << "Hello " << name << endl;
    return 0;
}
```

Setup workspace

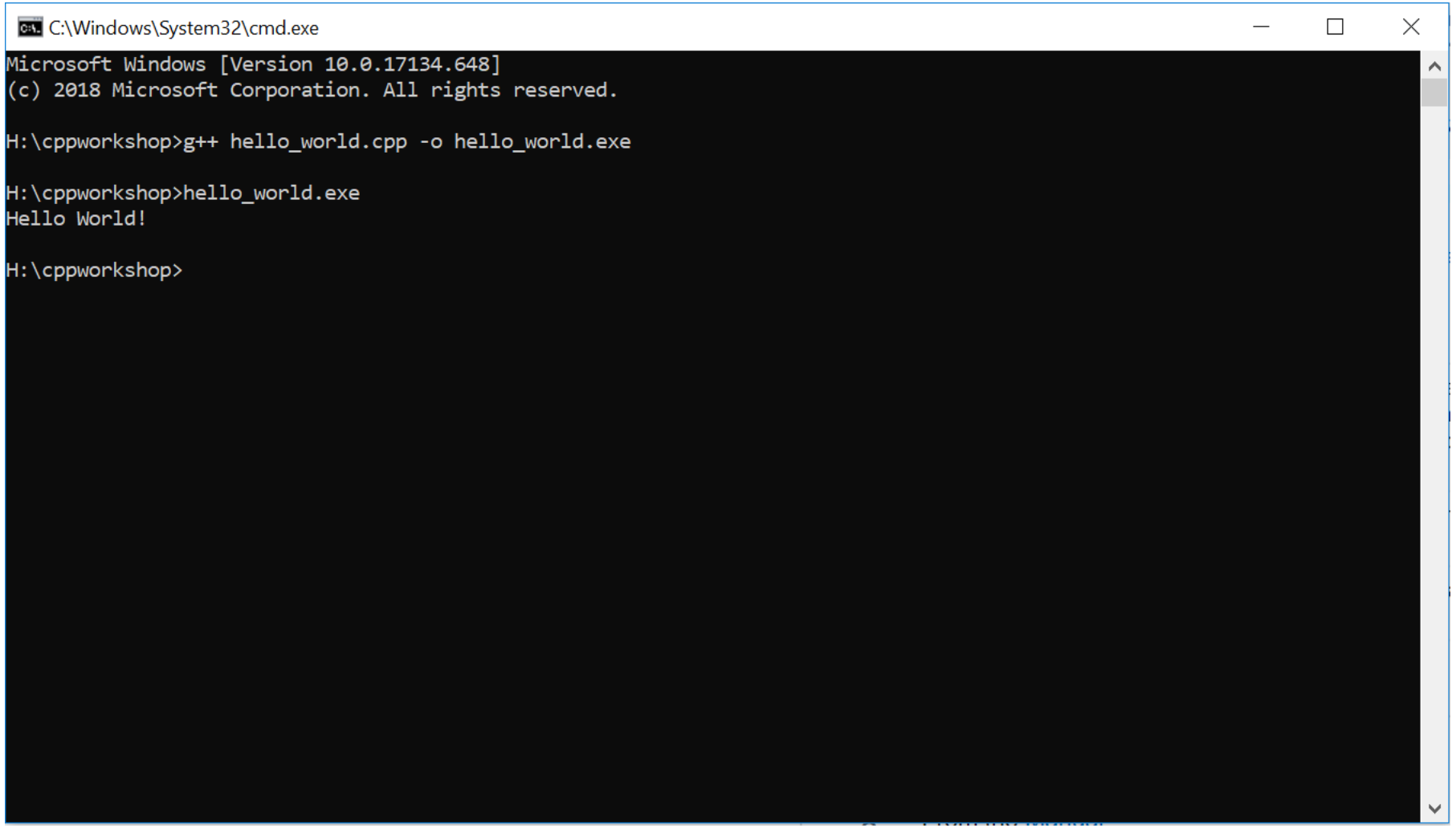
http://coliru.stacked-crooked.com/


File Edit Selection View Go Debug ... hello_world.cpp - Visual Stu...
hello_world.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6 cout << "Hello World!" << endl;
7 return 0;
8 }
0 0


cppworkshop
File Home Share View
This PC > psypp3 (\\spslinux) (H:) > cppworkshop
Quick access
Documents
Downloads
Pictures
cppworkshop
This PC
Network
Name Date modified Type Size
hello_world.cpp 27/03/2019 17:15 C++ Source 1 KB
how_to_compile_code 27/03/2019 17:16 File folder


cppworkshop
File Home Share View
cmd
Search cppworkshop
Quick access
Documents
Downloads
Pictures
cppworkshop
This PC
Network
Name Date modified Type Size
hello_world.cpp 27/03/2019 17:15 C++ Source 1 KB
how_to_compile_code 27/03/2019 17:16 File folder

compile with g++ (c++ on mac)



Practice!

https://www.hackerrank.com/