

PROGRAM COMPILATION, MAKEFILES ARRAYS AND STRUCTS

Problem Solving with Computers-I

<https://ucsb-cs16-wi17.github.io/>

C++

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

int main(){
    cout<<"Hola Facebook\n";
    return 0;
}
```

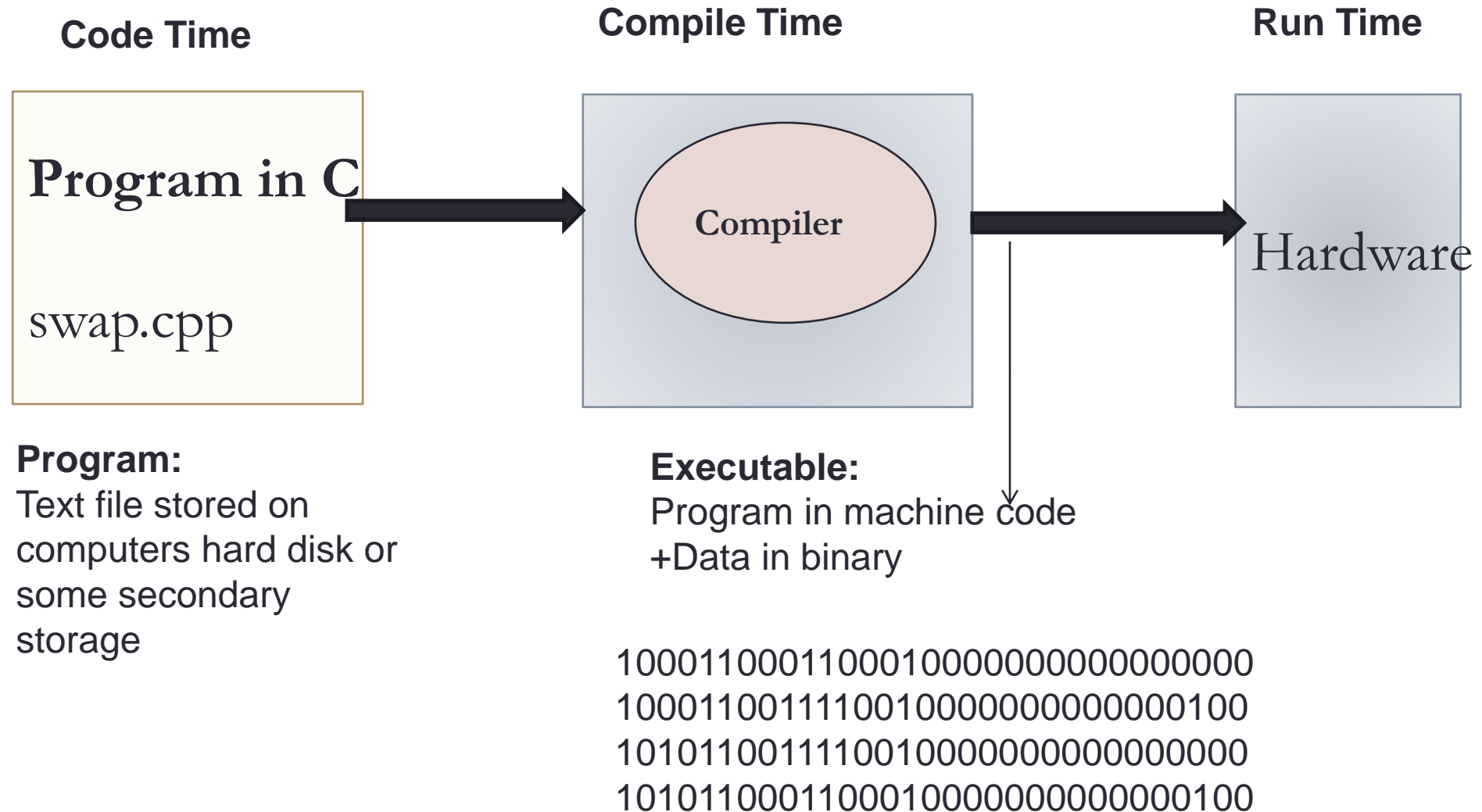


swap.cpp

```
void swapValue(int *x, int *y){  
    int tmp = *x;  
    *x = *y;  
    *y = tmp;  
}
```

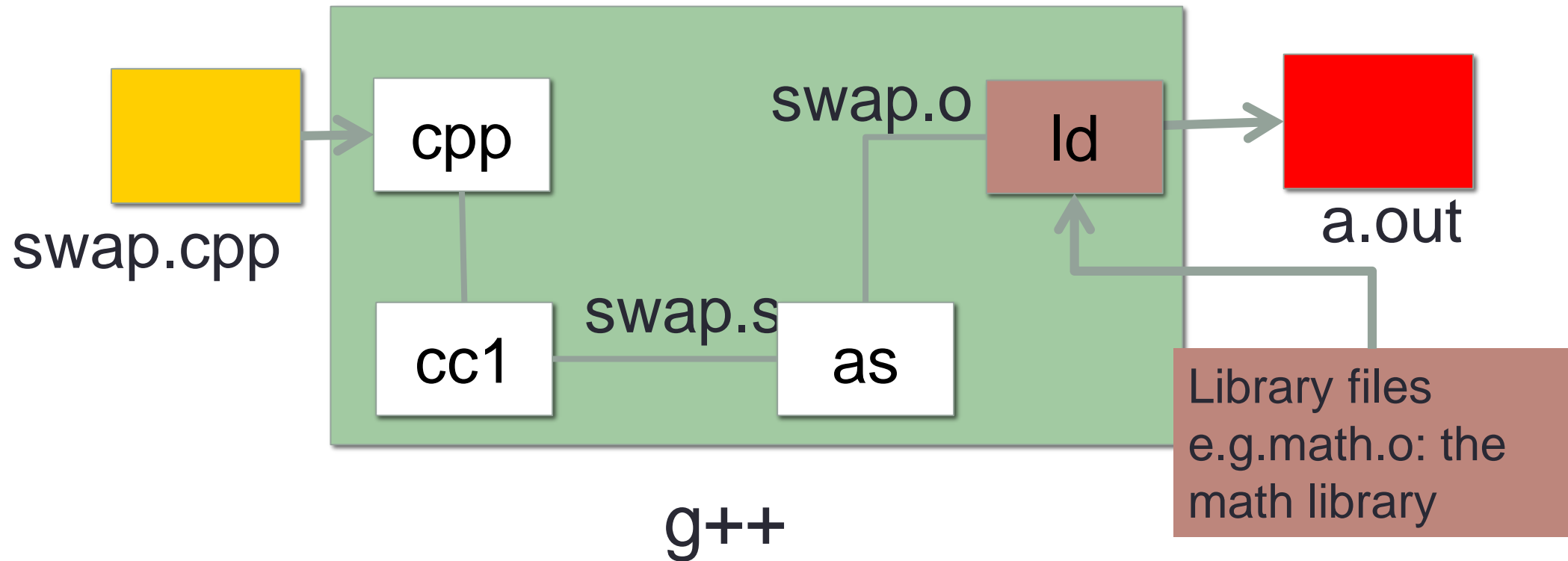
```
int main() {  
    int a=10, b=20;  
    swapValue( &a, &b);  
    cout<<a<<" "<<b<<endl;  
}
```

Steps in program translation



g++ is composed of a number of smaller programs

- Code written by others (libraries) can be included
- ld (linkage editor) merges one or more object files with the relevant libraries to produce a single executable



Steps in gcc

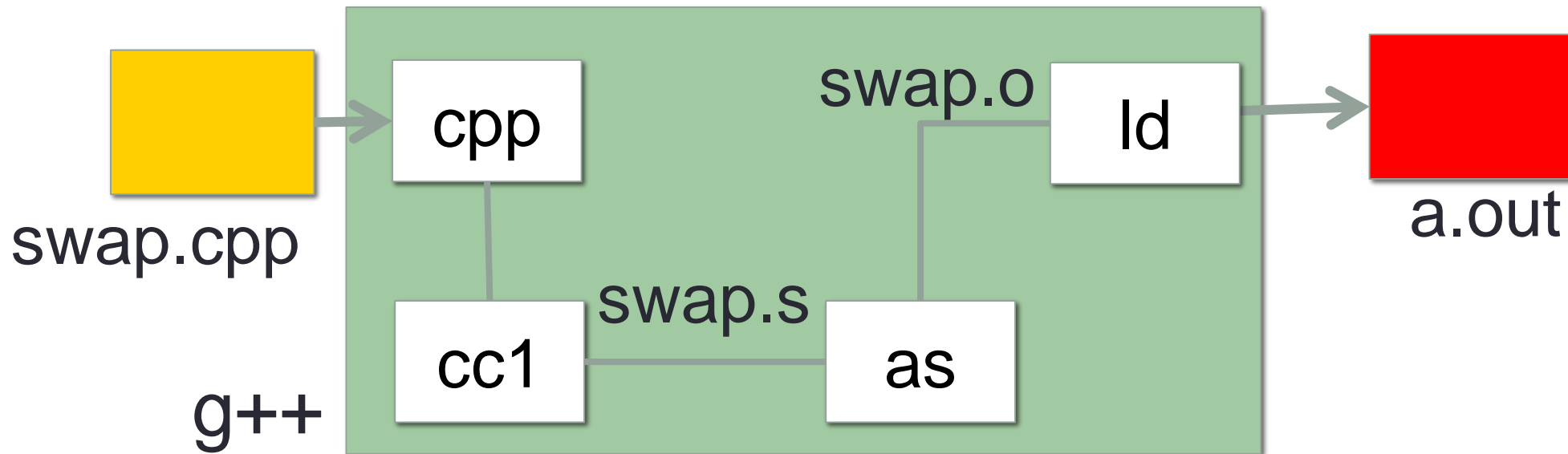
- Ask compiler to show temporary files:

```
$ g++ -S swap.cpp
```

```
$ g++ -c swap.o
```

```
$ g++ -o swap swap.cpp
```

```
$ g++ file1.o file2.o -o swap
```



Complex declarations in C/C++

How do we decipher declarations of this sort?

```
int **arr[];
```

Read

- * as “pointer to” (always on the left of identifier)
- [] as “array of” (always to the right of identifier)
- () as “function returning” (always to the right ...)

For more info see:

http://ieng9.ucsd.edu/~cs30x/rt_lt.rule.html

Complex declarations in C/C++

Right-Left Rule

```
int **arr [];
```

Step 1: Find the identifier

Step 2: Look at the symbols to the right of the identifier. Continue right until you run out of symbols *OR* hit a *right* parenthesis ")"

Step 3: Look at the symbol to the left of the identifier. If it is not one of the symbols '*', '(', '[' just say it. Otherwise, translate it into English using the table in the previous slide. Keep going left until you run out of symbols *OR* hit a *left* parenthesis "(".

Repeat steps 2 and 3 until you've formed your declaration.

Illegal combinations include:

[]() - cannot have an array of functions

()() - cannot have a function that returns a function

()[] - cannot have a function that returns an array

Complex declarations in C/C++

```
int i;  
int *i;  
int a[10];  
int f( );  
int **p;  
int (*p)[];  
int (*fp)( );  
int *p[];  
int af[]( );  
int *f();  
int fa()[];  
int ff()();  
int (**ppa)[];  
int (*apa[ ])[ ];
```


Next time

- Dynamic memory allocation