INTRO TO LAB02, MAKEFILES, REVIEW LOOPS/FUNCTIONS

Problem Solving with Computers-I

https://ucsb-cs16-sp17.github.io/

C++

GitHub

Clickers out – frequency AB

Demo

- Basics of code compilation in C++ (review)
- Makefiles (used to automate compilation of medium to large projects) consisting of many files
- We will start by using a makefile to compile just a single program
- Extend to the case where your program is split between multiple files
- By the end of this you should know what each of the following are and how they are used in program compilation
  - Header file (.h)
  - Source file (.cpp)
  - Object file (.o)
  - Executable
  - Makefile
  - Compile-time errors
  - Link-time errors

Steps in program translation

Code Time

Compile Time

Run Time

Program in C

swap.cpp

Compiler

Hardware
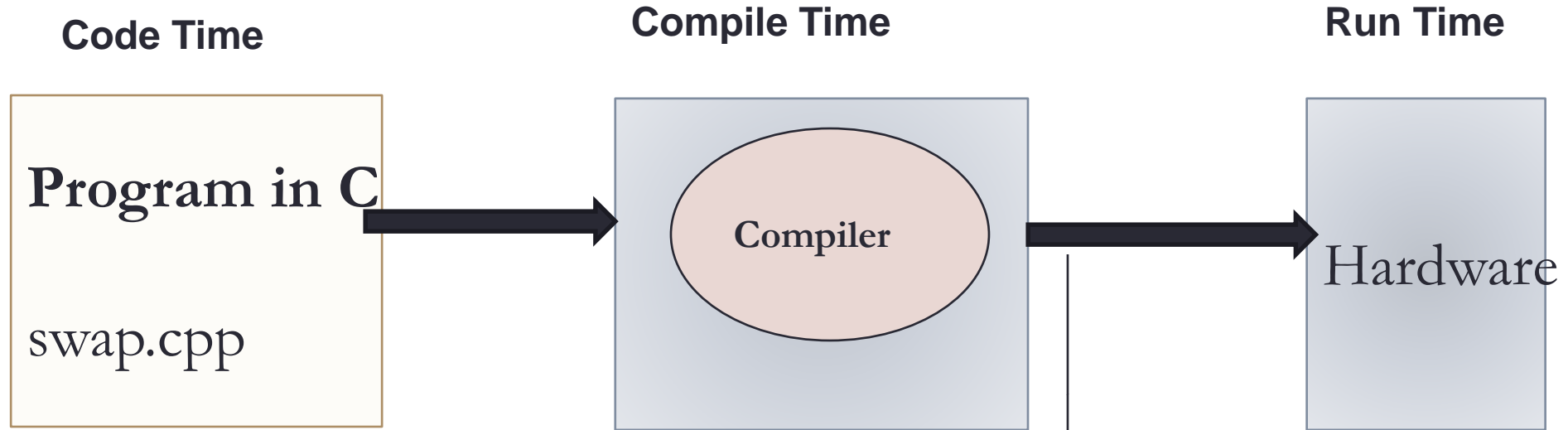
Program:
Text file stored on computers hard disk or some secondary storage

Executable:
Program in machine code
+Data in binary

100011000110001000000000000000000
100011001111001000000000000000100
101011001111001000000000000000000
101011000110001000000000000000100

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
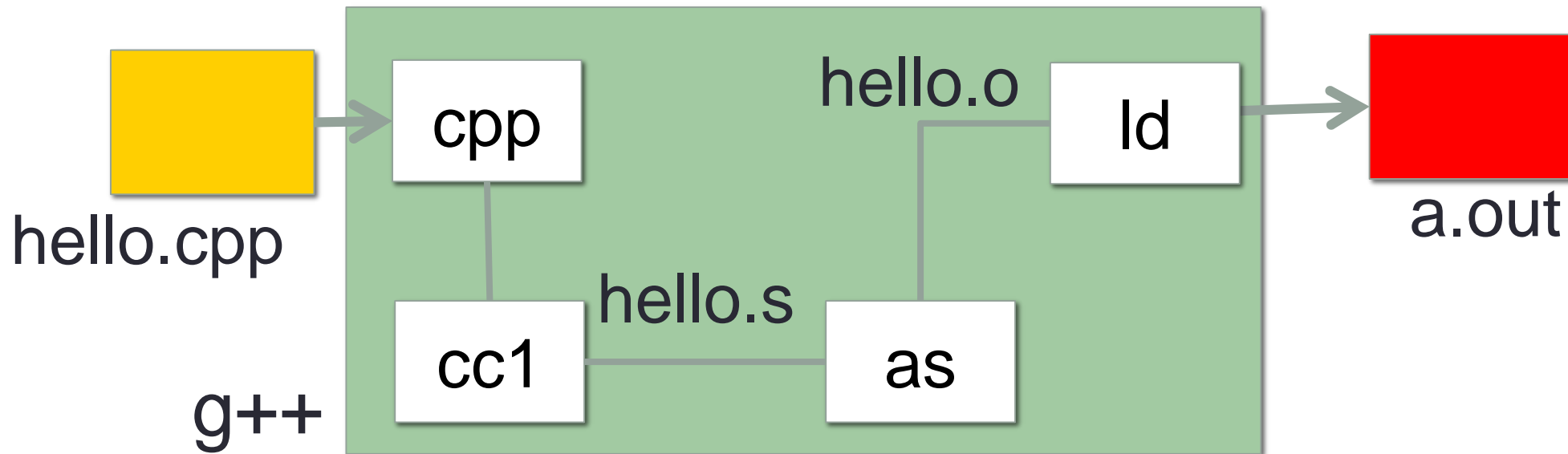
hello.cpp → cpp

cc1

hello.s

as

hello.o

ld → a.out

g++

Library files e.g.math.o: the math library
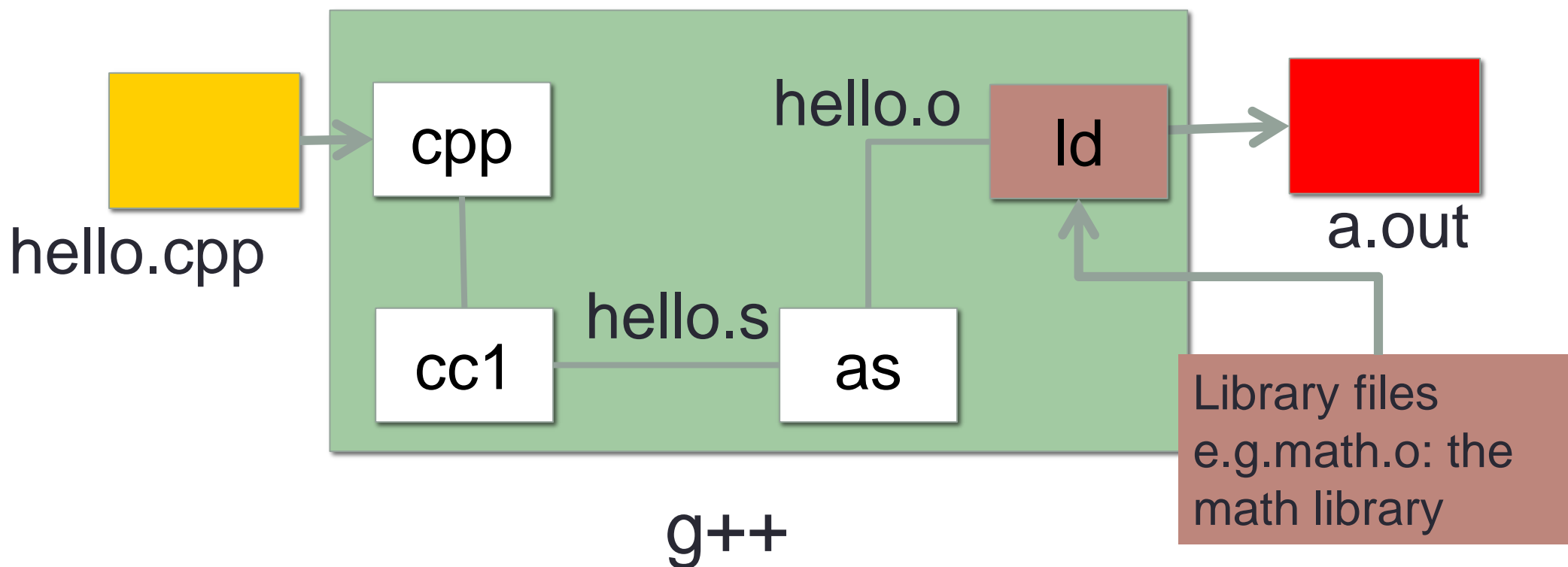
Steps in gcc

- Ask compiler to show temporary files:

$ g++ –S hello.cpp

$ g++ –c hello.o

$ g++ –o hello hello.cpp

$ g++ functions.o main.o –o myhello

Control Flow: while and do while loops

```
while(Boolean expression){
    //statement 1
    //statement 2
}


do{
    //statement 1
    //statement 2
  }while(Boolean expression);
```

Identify the code that is not equivalent to the other two?

Assume 'n' is an integer that has already been declared (may be positive or negative)

A.
```
for( int x = 0; x < n; x++ ) {
   cout<<x <<endl;
}
```

B.
```
int x = 0;
while(x < n) {
   cout<< x << endl;
   x++;
}
```

C.
```
int x = 0;
do{
   cout<< x<< endl;
   x++;
} while(x < n);
```

D. They are ALL equivalent

Infinite loops

```
for(int y=0;y<10;y--)
    cout<<"Print forever\n";
```

```
int y=0;
for(;;y++)
    cout<<"Print forever\n";
```

```
int y=0;
for(;y<10;);
    y++;
```

```
int y=0;
while(y<10)
    cout<<"Print forever\n";
```

```
int y=0;
while(y=2)
    y++;
```

Function call mechanics

What is the output of the following code

```cpp
int sum(int a, int b){
        cout<< a+b;
}
int main(){
        int result =0;
        int x =10, y =20;
        result = sum(x, y);
        cout<<result+30;
}
```

Function call mechanics

What is the output of the following code

```cpp
int sum(int a, int b){
        return a+b;
}
int main(){
        int result =0;
        int x =10, y =20;
        result = sum(x, y);
        cout<<result+30;
}
```

Function call mechanics

What is the output of the following code

```cpp
int sum(int a, int b){
        int result= a+b;
        exit(0);
}

int main(){
        int result =0;
        int x =10, y =20;
        result = sum(x, y);
        cout<<result+30;
}
```

for loop OR while loop? Which one should you use?

```cpp
for (int i = 0; i < 15; i++){
    cout << i << endl ;
}


int j =0;
int n;
while(j < 15){
    cout << "Enter a number" << endl ;
    cin>>n;
    j = j+n;
    cout<<"Current value of j is:"<<endl;
}
```

Next time

- File IO
- Intro to lab03