

Programming in C

C code tool chain

- Pre-processor (`gcc -E`)
- Compiler (`gcc -c`)
- Linker (`ld`, but invoked via `gcc` for us)
- Loader (`ld-linux.so`)

Example

- ▶ Open a text editor (geany) and type in the following program:

```
#include <stdio.h>

int main (void)
{
    printf ("Hello World");
    return 0;
}
```

- ▶ Save it as (helloworld.c) in a folder called (HelloWorld) in ~/Documents/CS300

Example

- ▶ Now, open a terminal and navigate to the folder HelloWorld that you just created
- ▶ Type: `gcc -c helloworld.c`
- ▶ List the file contents. What file has been created?
- ▶ Type: `gcc helloworld.o`
- ▶ List the file contents. What file has been created?
- ▶ Type: `./a.out`
- ▶ What happens?

Loops, Arrays, Pointer Review

- Using `scp`, copy the file `examstats.c` from `/home/CS300Public/2017` on zeus into your CS300 folder
- Open the file in geany, Compile, Build, and Execute
- Let's go over the code

Questions on the Previous Program

- ▶ What is the purpose of the **#define** statement and who does it create work for:
 - ▶ The pre-processor
 - ▶ The compiler
 - ▶ The linker
 - ▶ The loader
- ▶ What is the purpose of using **static** in a function prototype?
- ▶ Explain `scanf`
- ▶ What does “%6d” mean?