

c

WHY LEARN C?

- help develop better understanding of what higher (and lower) level languages are doing
- help understand memory management
- system programming - interact with the OS
- to pass this class :)

C VS JAVA

- Both are compiled, but to different things:
 - C -> compiled to assembly language
 - Java -> compiled to Java byte code, executed by JVM
- Portability:
 - Java = more portable, compiled byte code runs the same on same JVM version
 - C = less portable, compiled code is architecture dependent

C VS JAVA (CONT.)

- Speed
 - C often faster than Java
 - Java can be fast (JIT can make optimizations)

COMPILED VS INTERPRETED

- Compiled:
 - C (`gcc program_file.c`)
 - Java (`javac program_file.java`)
- Interpreted: can simply run without explicitly compiling
 - Bash (`bash name_of_program.sh`)
 - Python (`python name_of_program.py`)
 - Awk (`gawk name_of_program.awk`)

GETTING STARTED WITH C

Hello World Program

```
#include <stdio.h>
int main(void) {
    printf("Hello World\n");
    return 0;
}
```

C PREPROCESSOR

- Performs source code substitution
- # indicates preprocessing directive
- `include` - replaces line with contents of file
 - typically use to include header files (we'll talk about what these mean later)
 - kind of like "importing"
- `define PI 3.14159`
 - everywhere `PI` occurs, replace with value
 - typically use all caps

RUNNING C CODE

Compiling

```
gcc program_file.c
```

```
gcc -o output_name program_file.c
```

Running

```
./a.out
```

```
./output_name
```


:(DON'T DO THIS

```
gcc -o project.c project.c
```