COMP26020 Programming Languages and Paradigms -- Part 1

Introduction to C

C: Origin and Popularity

- C was designed in the 70s by Dennis Ritchie
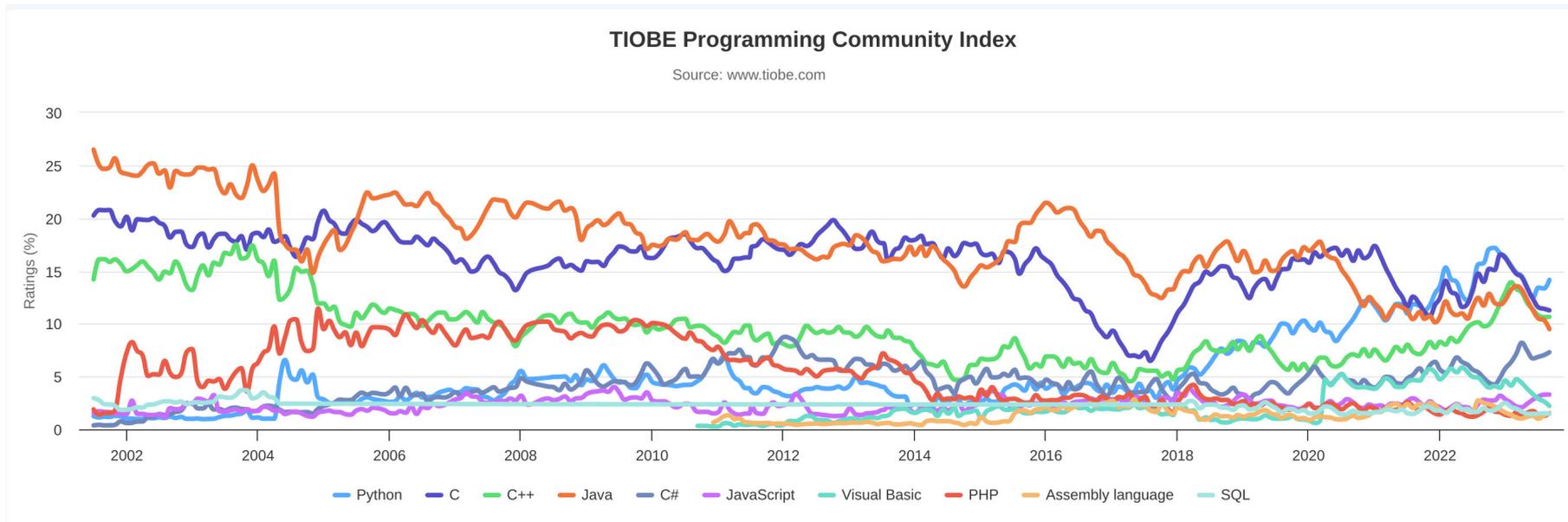
Still very popular today!

C: Characteristics

- **Pros:**
  - Syntax
  - Low level
  - Fast
  - Controlled memory footprint

C: Characteristics

- **Pros:**
  - Syntax
  - Low level
  - Fast
  - Controlled memory footprint

- **Cons**: large area for making mistakes
  - Lack of memory safety, risks of undefined behaviour at runtime

C: Characteristics

- **Pros:**
  - Syntax
  - Low level
  - Fast
  - Controlled memory footprint

- **Cons**: large area for making mistakes
  - Lack of memory safety, risks of undefined behaviour at runtime

Still extensively used in: systems software, high performance computing, embedded systems, etc.

Popular Software Written in C
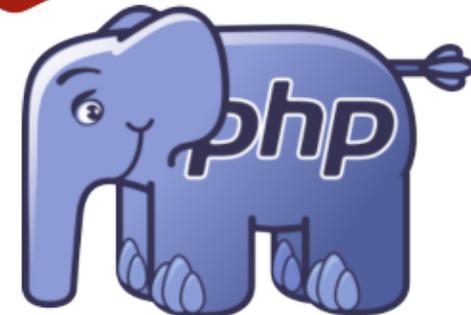
Hello World

```c
#include <stdio.h>

int main() {

    printf("hello, world!\n");

    return 0;
}
```

Assuming the code is in `hello-world.c`, to compile and run on the Linux command line:

```
$ gcc hello-world.c -o hello
$ ./hello
hello, world!
```

Hello World

```c
#include <stdio.h>

int main() {

    printf("hello, world!\n");

    return 0;
}
```

Hello World

```c
#include <stdio.h>

int main() {

    printf("hello, world!\n");

    return 0;
}
```

- `#include` to get access to functions from external libraries

Hello World

```c
#include <stdio.h>

int main() {

    printf("hello, world!\n");

    return 0;
}
```

- `#include` to get access to functions from external libraries
- `main` will run first when the program is executed

Hello World

```c
#include <stdio.h>

int main() {

    printf("hello, world!\n");

    return 0;
}
```

- `#include` to get access to functions from external libraries
- `main` will run first when the program is executed
  - It returns an integer (`int`) and takes no parameters

Hello World

```c
#include <stdio.h>

int main() {

    printf("hello, world!\n");

    return 0;
}
```

- `#include` to get access to functions from external libraries
- `main` will run first when the program is executed
    - It returns an integer (`int`) and takes no parameters
    - Contains two statements, in C statements ends with `;`

Hello World

```c
#include <stdio.h>

int main() {

    printf("hello, world!\n");

    return 0;
}
```

- `#include` to get access to functions from external libraries
- `main` will run first when the program is executed
    - It returns an integer (`int`) and takes no parameters
    - Contains two statements, in C statements ends with `;`
- `printf` used to print text on the *standard output* (console)

Hello World

```c
#include <stdio.h>

int main() {

    printf("hello, world!\n");

    return 0;
}
```

- `#include` to get access to functions from external libraries
- `main` will run first when the program is executed
    - It returns an integer (`int`) and takes no parameters
    - Contains two statements, in C statements ends with `;`
- `printf` used to print text on the *standard output* (console)
- `return` to go back to the calling context, and exit if returning from `main`

Compilation

hello-world.c

```
#include <st
int main() {
    printf("
    return 0
}
```

Compiler (gcc)

hello

```
0111010101
0101110111
1011100111
0111010101
0101110111
0101110111
1011100111
```

Source file
(text)

Executable program
(binary)
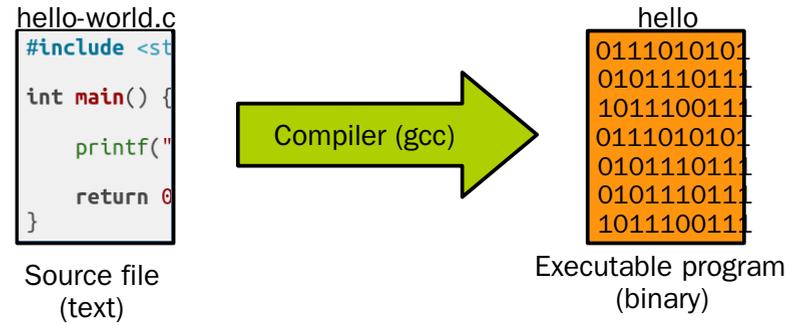
```
$ gcc hello-world.c -o hello
```

Compilation

```
#include <stdio.h>

nt main() {
    printf("hello, world!\n");

    return 0;
}
```

Compilation

```
#include <stdio.h>

nt main() {
    printf("hello, world!\n");

    return 0;
}
```

```
$ gcc hello-world.c -o hello
hello-world.c:3:1: error: unknown type name 'nt'; did you mean 'int'?
    3 | nt main() {
      | ^~
      | int
```

Compilation

```
#include <stdio.h>

nt main() {
    printf("hello, world!\n");

    return 0;
}
```

```
$ gcc hello-world.c -o hello
hello-world.c:3:1: error: unknown type name 'nt'; did you mean 'int'?
    3 | nt main() {
      | ^~
      | int
```

Multiple issues? fix them in the order they are reported

Compilation: Warning and Errors

- Errors are unrecoverable
- Warnings *may* indicate a problem

```c
#include <stdio.h>

int main() {
    int x;
    printf("hello, world!\n");

    return 0;
}
```

```
gcc -Wall hello-world.c -o hello
hello-world.c: In function 'main':
hello-world.c:4:9: warning: unused variable 'x' [-Wunused-variable]
     int x;
         ^
```

Compilation: Warning and Errors

- Errors are unrecoverable
- Warnings *may* indicate a problem

```c
#include <stdio.h>

int main() {
    int x;
    printf("hello, world!\n");

    return 0;
}
```

```
gcc -Wall hello-world.c -o hello
hello-world.c: In function 'main':
hello-world.c:4:9: warning: unused variable 'x' [-Wunused-variable]
    int x;
        ^
```

- **Warnings almost always indicate a problem and you should fix them**
- Some warnings (picky ones) are disabled by default, enable them with the `-Wall -pedantic` flags

Comments

```c
/* listing2.c, illustrate the use of comments */

#include <stdio.h> // necessary to get access to printf

/* this function simply prints out 'hello world' and returns */
int main() {

    printf("hello, world!\n"); // here we print ...

    /* the line below will not be compiled: */
    // printf("goodbye!\n");

    return 0;                  /* ... and return */
}
```

03-c-introduction/comments.c

```c
/* style 1 */
// style 2
```

<u>Use comments</u> to explain what your code does, uncommented code is hard to understand

Summary

- Brief intro to C
- Hello world
- Compilation process
- Comments

Feedback form: https://bit.ly/3yzIVls