COMP1511 Tutorial 7

pointers | struct pointers | eof loops

pointers revision

Pointer Operations

`int *ptr` - Declare an integer pointer called `ptr`

`&num` - Give me the address of variable `num`

`*ptr` - Give me the variable at the address stored in `ptr` (dereferencing)

struct pointers

Do I use `.` or `->`?

With a regular struct variable (e.g. a `struct person` named `student1`), we would the `.` (dot) operator to access a field (e.g. `student1.name`)

If we had a pointer to a struct, we would need to dereference the pointer to get to the struct and then access its field: `(*student1_pointer).name`

We use the `->` operator to make this neater: `student1_pointer->name`

scanf, fgets and eof

scanf or fgets?

`scanf` is useful when you need to operate character by character and don't need to store a whole string to use later

`fgets` is very convenient for scanning in a whole line of input and storing the whole string, but does require declaring a char array of a certain length beforehand

Keep in mind: `scanf` returns EOF (an integer) when EOF is reached, `fgets` returns NULL (a pointer) when EOF is reached

side notes

Pointers: declaring vs dereferencing

The asterisk (*) is used for 2 key pointer operations:

Declaring, e.g. `int *ptr;`

Dereferencing, e.g. `*ptr = 5`

Rule of thumb: if the asterisk has a variable type before it (e.g. `int *`, `char *`) it's a pointer declaration, otherwise it's dereferencing a pointer

EOF (end-of-file)

When a file stream has no more input to give to a program, the program will receive the EOF signal to tell it that no more input will be given.

The file stream we've been using has been `stdin` (standard input), which reads from the terminal, where pressing `ctrl-d` sends the EOF signal.

Autotests use text files to input values into our programs, and EOF is sent automatically after all the data in the file has been inputted.

When your program receives EOF, any `scanf`'s and `fgets`' will essentially be skipped (otherwise your program would be left waiting forever for input that it will never receive!