P2T

C

Lecture 1:

- C is medium-level, structured, procedural, imperative programmming language
- close to low-level (assembly language), but does not map the functions one-to-one
- we can use functions to call parts of code multiple times

C program Structure

```
#include <stdio.h> # starts with this, tells the linker, where to find needed libraries

int main() # each c program has to have main function, but there can be other functions as w
{
    int i, j; # these are declarations and statement
    i = 3;
    j = i * 2;
    printf("%d doubled is %d\n",i,j); # function-call statement
    return 0;
}
```

Running code

- to run C program, we usually use `gcc -o test myProg.c`, where test is the output file
- then we use `./test`

Types

- integers - numbers
  - short - 2 bytes
  - int - 4 bytes
  - long - 8 bytes
  - each type has also signed and unsigned (positive only) subtype
- floating-point numbers
  - float - 4 bytes
  - double - 8 bytes
  - long double - 16 bytes
- other types:
  - characters
    - single caracters of text
    - only single quotes `'...'`
    - they use ASCII

        * to print special characters, such as quotes, we have to escape them
            - `\'`
    - void

Computer memory

- bit is the smallest stored information - 0/1
- byte is more useful part of mmemory, as it is 8 bits
- complete unit of memory is called a word, which consists of one or more bytes
- memory can be expressed as a huge number of rows, where each row is a word

Operators

- they let us modify the variables
- LECTURE NOTES FOR THESE
- Arithmetic Operators
    - they work on any data type that is numeric
    - division of two integers is rounded down
    - if we mix floats with ints, the result will be float
- Logical Operators
    - operators for boolean operations `&&` - and, `||` - or, `!` - not
- Others
    - `sizeof()` - number of bytes that the object uses
    - it is not a function, it is an operator