C-Week-1 Notes

L1 Writing First C Prog.

- idea - what problem has to be solved (prelim docs, specificaⁿ)
- architecture - memory layout, data struc., flowcharts (paper design)
- Coding - files with instrucⁿ ≡ prog. (editor, IDE)
- Running & Debugging - prog. → code, observation (compiler, shell)
- use use explit
- ```
  #include <stdio.h>
  int main(void){
     printf("Hello World\n");
     return 0;
  }
  ```

L2 Struc. of a C Prog.

```
#include <stdio.h>   → include header
#define N 10         → macro processor
```
```

```
int addNums(int *a, int numvals){
  int i;
  int sum;
  sum = 0;
  for (i=0; i<numvals; i++){
     sum = sum + a[i];
  }
  return sum;
}
int main(void){
  int a[N] = {_ _ _ _ .};
  printf("%d", addNums(a,N))
  return 0;
}
```

func^n → params

a[i] → expression

sum; → statement

- comments —
  1. //
  2. /*
     */

- grammer - rules of a lang., syntax (1st step checked by a compiler), semantics - meaning.
- style - aesthetics (diff. to quantify), indenta^n, naming & structure.

L3 Variables in C

- Memory/Register loca^n → Variable
- name for some data stored at some address.
- types – int, short, char, float, double ...
- interpreta^n of bytes in memory, once declared, can't Δ.
- assignment – set or update a value, corresponding memory loca^n.
- loca^n can't be specified by the prog.
- 0 → False, non-0 → True

L4 Operators in C

- computa^n, updating values, closely related to func^{ns}, used to construct expressions.
- specific syntax rules on usage.
- aethematic → +, −, *, /, %

- Type-dep. → 5/2 = 2 (int)
  5/2.0 = 2.5 (float)

- Ternary op. (if-else)
  ((x>10) ? -3 : +5
  ↳useful for concise & compact code, use with care.

- i = 3
  i++ ⇒ 4
  ++i ⇒ 5
  j = ++i ⇒ i = 6, j = 6
  j = i++ ⇒ j = 6, i = 7

- a = (x=10, y=20)
  ⇒ a = 20

- Conditions
  Logical ⇒ a==b, >, <, >=, <=
  Combina^n ⇒ ||, &&

- size (in bytes) → occupied by a var. or data type.