CS2100 - L02 - Overview of C

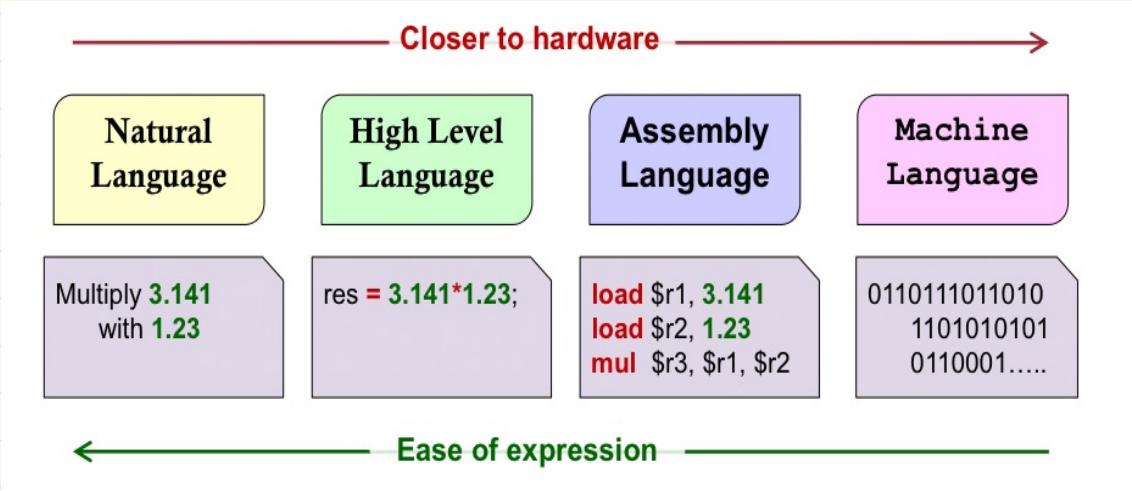
2.1 - Programming Languages

2.2 - History of C

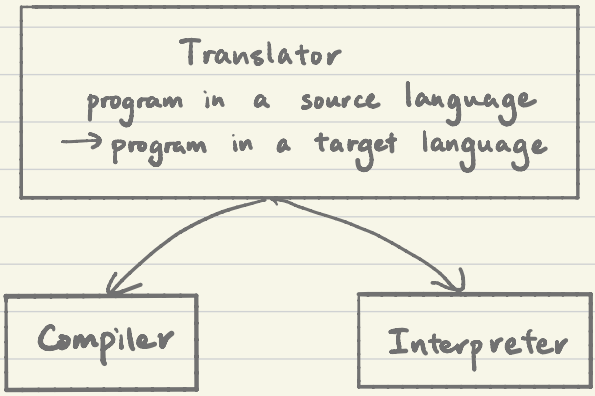
2.3 - Basic C Elements

2.4 - C Reference

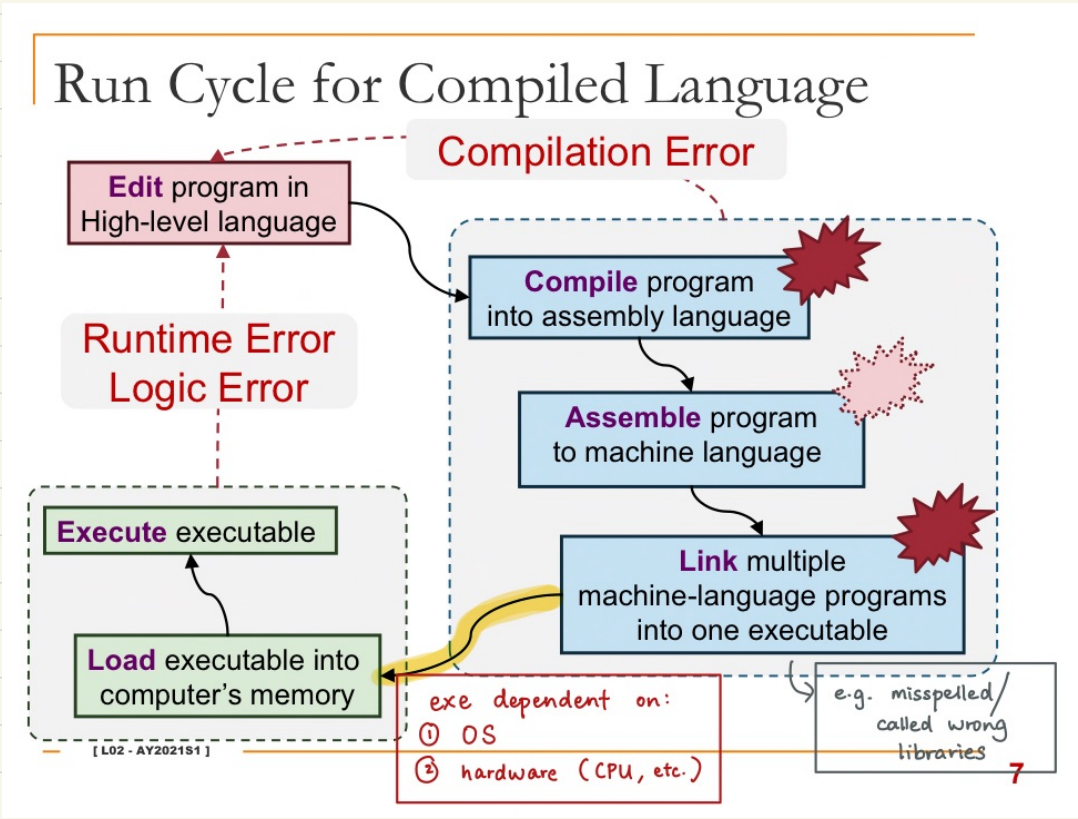
2.1 - Programming Languages



Translators

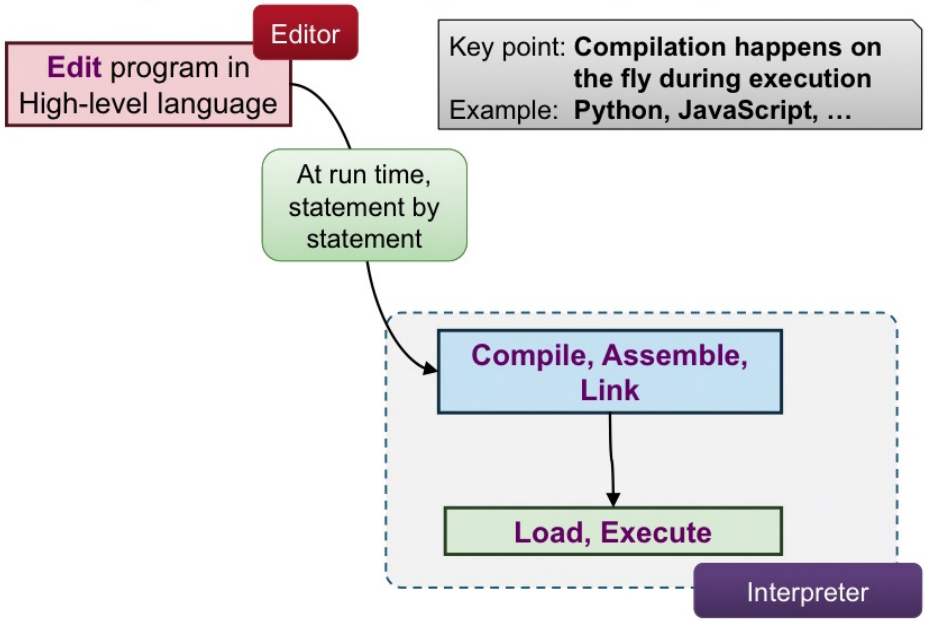


Run Cycle for Compiled Languages

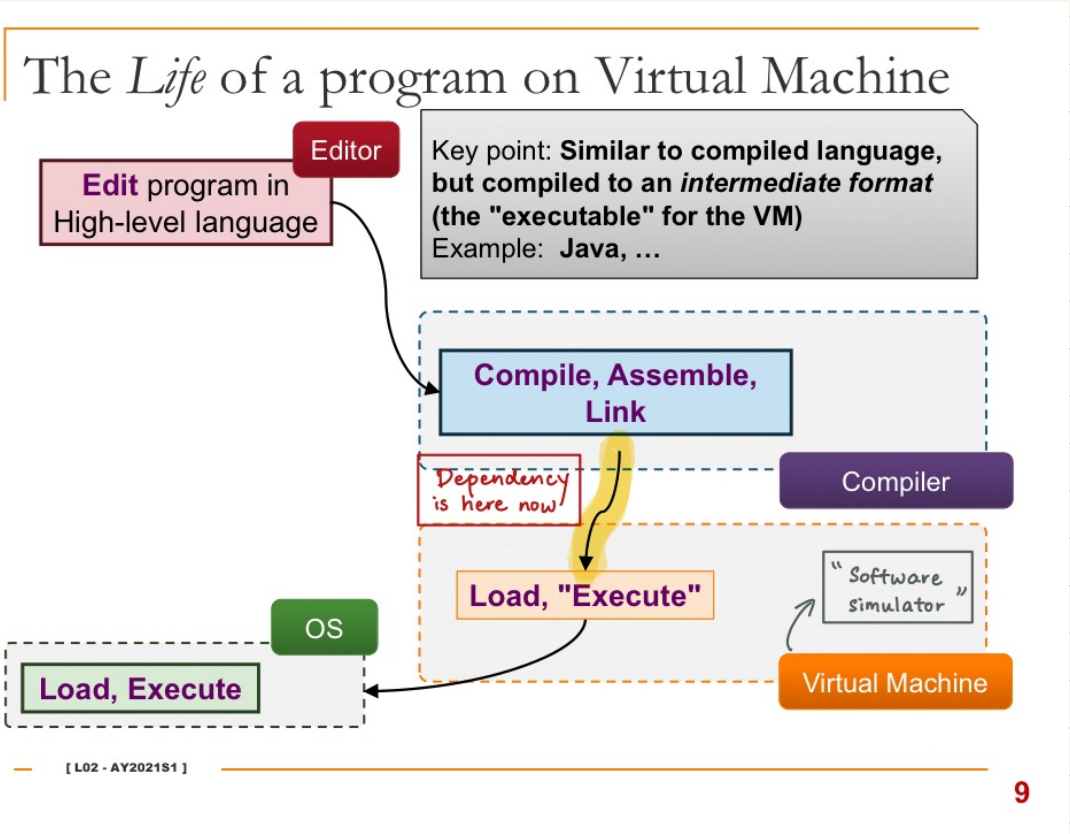


Interpreted Programs

The *Life* of a **interpreted** program



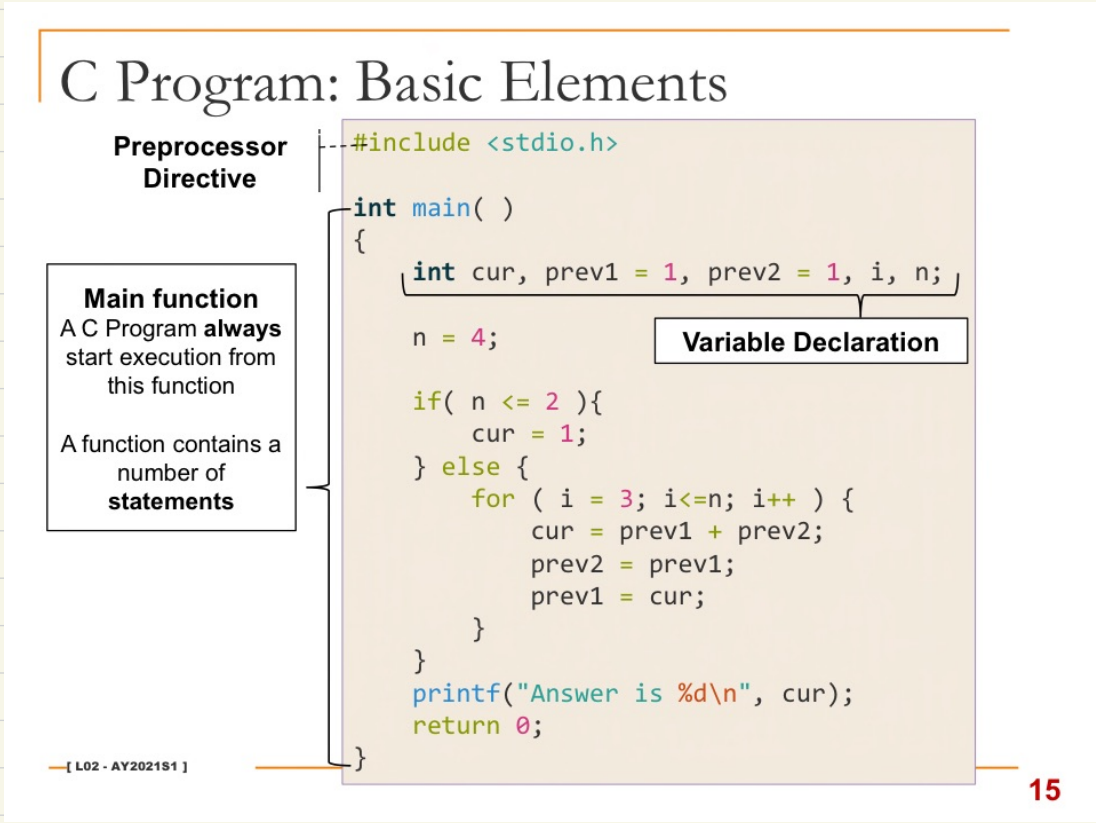
Programs on Virtual Machines



2.2 - History of C

2.3 - Basic C Elements

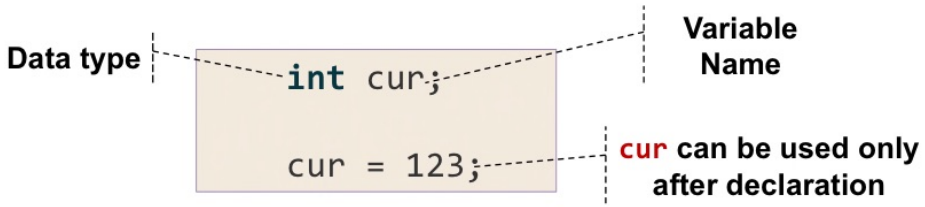
C : Basic Elements



C: Variables

C Variable

- Perhaps the first stumbling block for Python / JavaScript programmer
- **C Variable**:
  1. Must be **declared** before use
  2. Declaration must includes **data type**
  3. Do **not** have an initial value (i.e. cannot assume to be zero if uninitialized)



C: Data Types

C Variable – **Data Type**

- Data type dictates:
  - [Obvious] The type of values stored in the variable
  - [Not obvious] The **range** of values that can be stored
  - [Not obvious] The **size** of the variable in memory
- Common Data Types: No need to memorise table ☺

Data Type	32-bit Processor	64-bit Processor
`int`	Size: 4 bytes (32 bits) Range: -2^{31} to $2^{31}-1$	Size: 8 bytes (64 bits) Range: -2^{63} to $2^{63}-1$
`float`	Size: 4 bytes (32 bits)	Size: 8 bytes (64 bits)
`double`	Size: 8 bytes (64 bits)	Size: 16 bytes (128 bits)
`char`	Size: 1 byte (8 bit) Range: -2^{7} to 2^{7}	

C : Limitations

① Range limitations

```c
int i = 1;

while (i > 0) {
   i++;    //i.e. i = i + 1;
}
printf("What?!\n");

return 0;
```

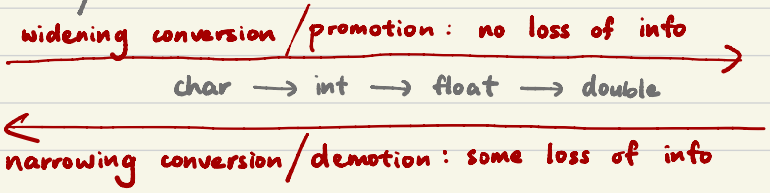
② Accuracy limitations

```c
double d = 0;
int i;

for (i = 0; i < 10000; i++){
    d = d + 0.1;
}

printf("d is %.12f\n", d);

return 0;
```

More in L03

C: Data Type Conversion

- Hierarchy:



C: Data Type Conversion Rules

① Operations: A op B → C (implicit)

- "lower" data types automatically promoted to match "higher" data type
- 2.0 / 5.0 → 0.4

  2.0 / 5 → 0.4 (5 promoted to 5.0)

② Assignment: A = B (implicit)

- B promoted / demoted to match A
- int myInt = 0.4; // myIntstores 0

  double myDouble = 5 // myDouble stores 5.0

③ Type casting: (data type) (explicit)

- (double) 2 / 5 → 0.4 (2 promoted to 2.0)

2.4 - C Reference