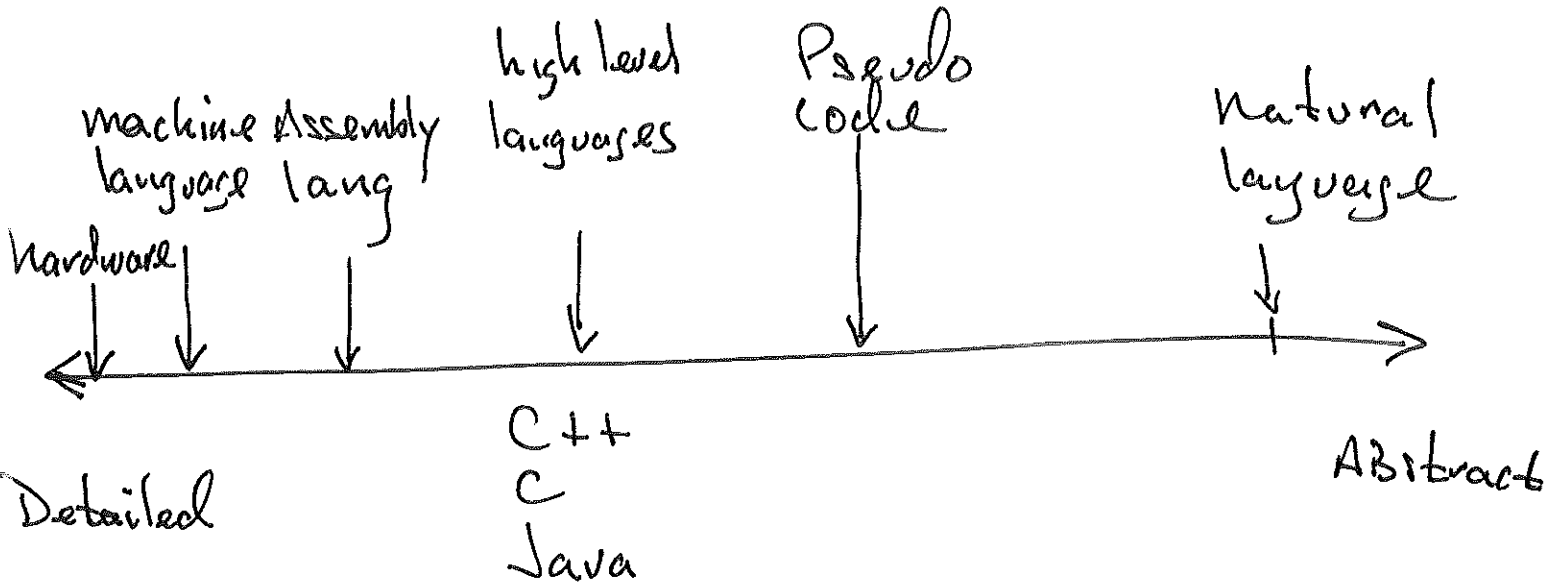


CNAPS ID 11-16-10

11

Higher level languages

language Abstractness Continuum



Pseudo code



C++ Program (source file) Prog.cpp



Assembly lang. Prog.



machine lang. Prog.



Complete mach. lang. (executable) Program



Hardware

Automated by g++

%g++ -o prog Prog.cpp

or

%g++ Prog.cpp

object file: a.out

Library Code
→
linker

load
Run

Automated by typing
name of executable

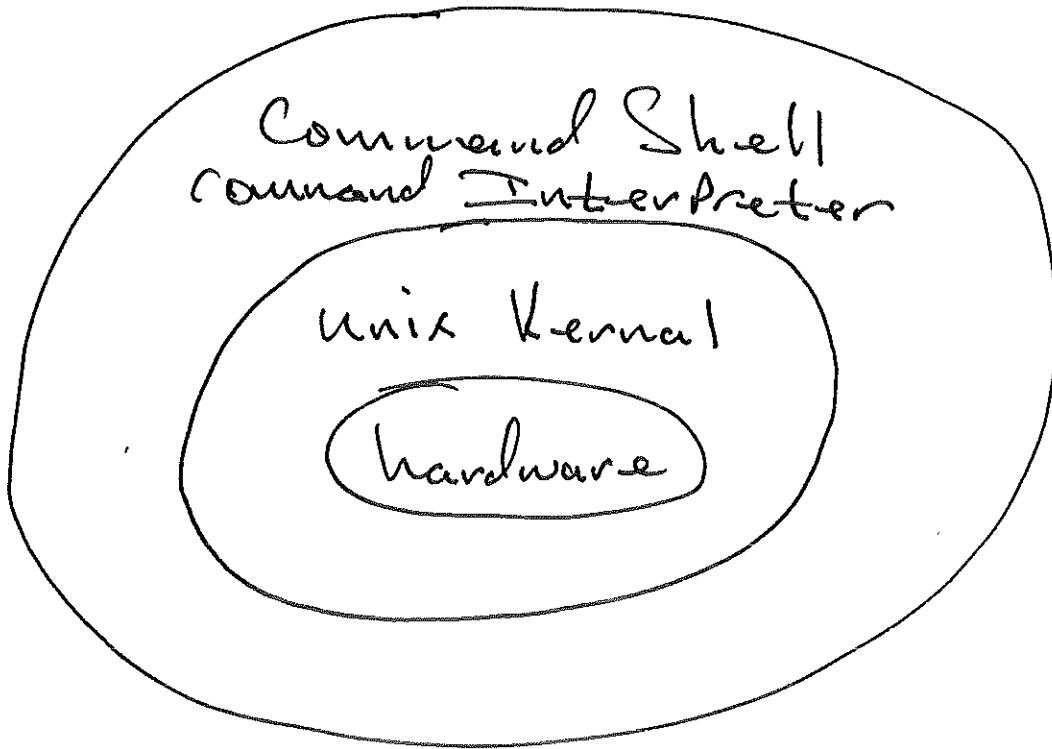
%prog

or

%a.out

Unix Environment

3



Command Shell

csk

tcsh

bash

ksh

sh

} common favorites

Form of a C++ Program

```

//
// comments
//
#include <iostream>
using namespace std;
#define .....

```

Preprocessor commands

Define Constant macros

classes } not covered
 functions }

```

int main(void) {
    // variable declarations
    // executable statements
    return 0;
}

```

classes } not covered
 functions }

Constant Macro:

5

#define PI 3.14

does a literal Text substitution

Variable Declaration

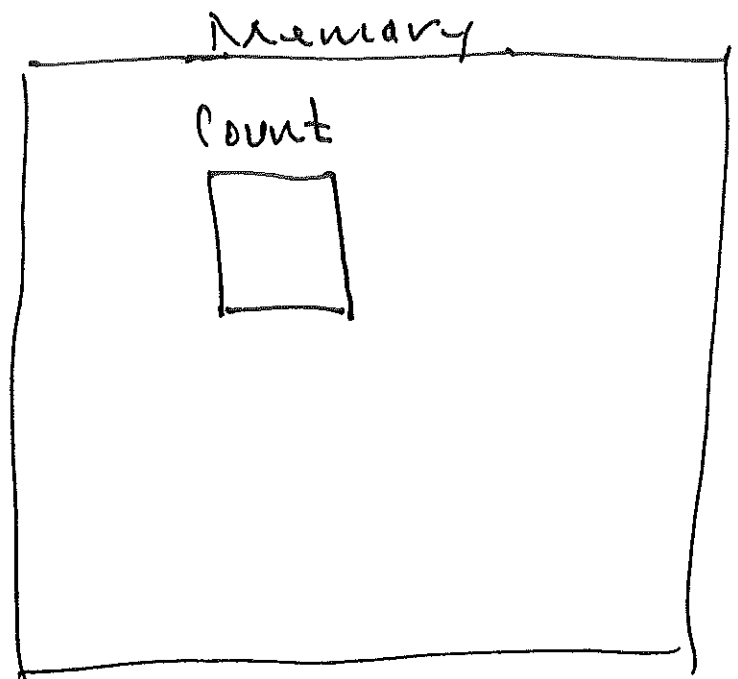
Define symbolic names for memory.

form:

data_type identifier, identifier, ... ;

Ex. int count;

↗
Data type for
Signed integer
Typically 4 bytes long
1 Byte = 8 Bits
4 Bytes = 32 Bits



Ex. double sum, average;

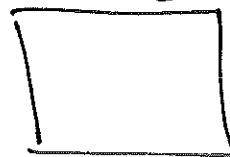
floating point number.

64 Bits
(8 Bytes)

sum



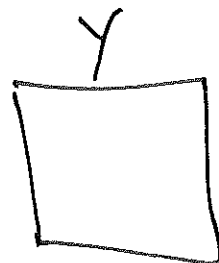
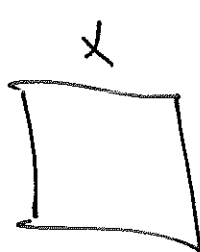
average



Ex. float x, y;

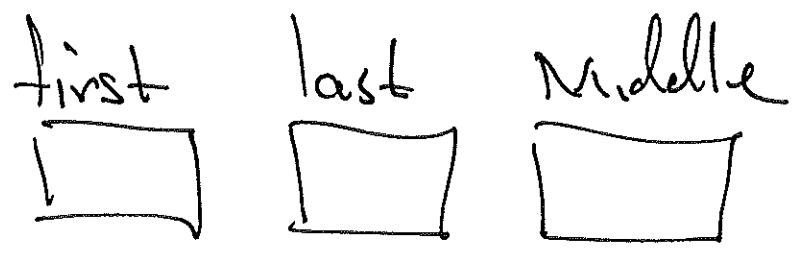
floating point number

32 Bits
(4 Bytes)

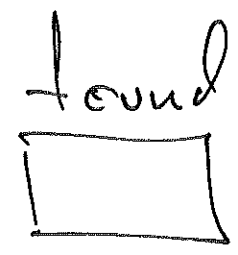


Ex. char first, last, middle;

Single ASCII
Character
1 BYTE (8 bits)



Ex. bool found;



Boolean: two values
true
false

Rules for Valid Identifiers

18

- consist only of
 - letters: a, b, ..., z, A, B, ..., Z
 - digits: 0, 1, ..., 9
 - underscores _
- may not
 - begin with a digit
 - be a reserved word

Ex. Valid

happy, Happy, happy1, happy_happy

Ex. Invalid

1happy, return, int, "happy"

$$16 + 8 + 1 \quad \boxed{\text{Example}}$$

$$= -25.3125$$

$$\frac{1}{4} = .25 \quad \boxed{9}$$

$$\frac{1}{16} = .0625$$

$$.3125$$

$$= -[11001.0101]_2$$

$$= -[.110010101]_2 \cdot 2^{[101]_2}$$

SUPPOSE 10 Bits mantissa
 5, 6 Bits EXP

