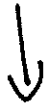


Pseudo-code



C++ Prog. (source file) myProg.cpp



Assem. lang



Machine lang.



Complete Mach. lang. (object file)

Program



Hardware

Library code (linker) →

} Automated in Unix by g++
% g++ -o myProg myProg.cpp

or

% g++ myProg.cpp
object file: a.out

} Automated in Unix by typing name of executable

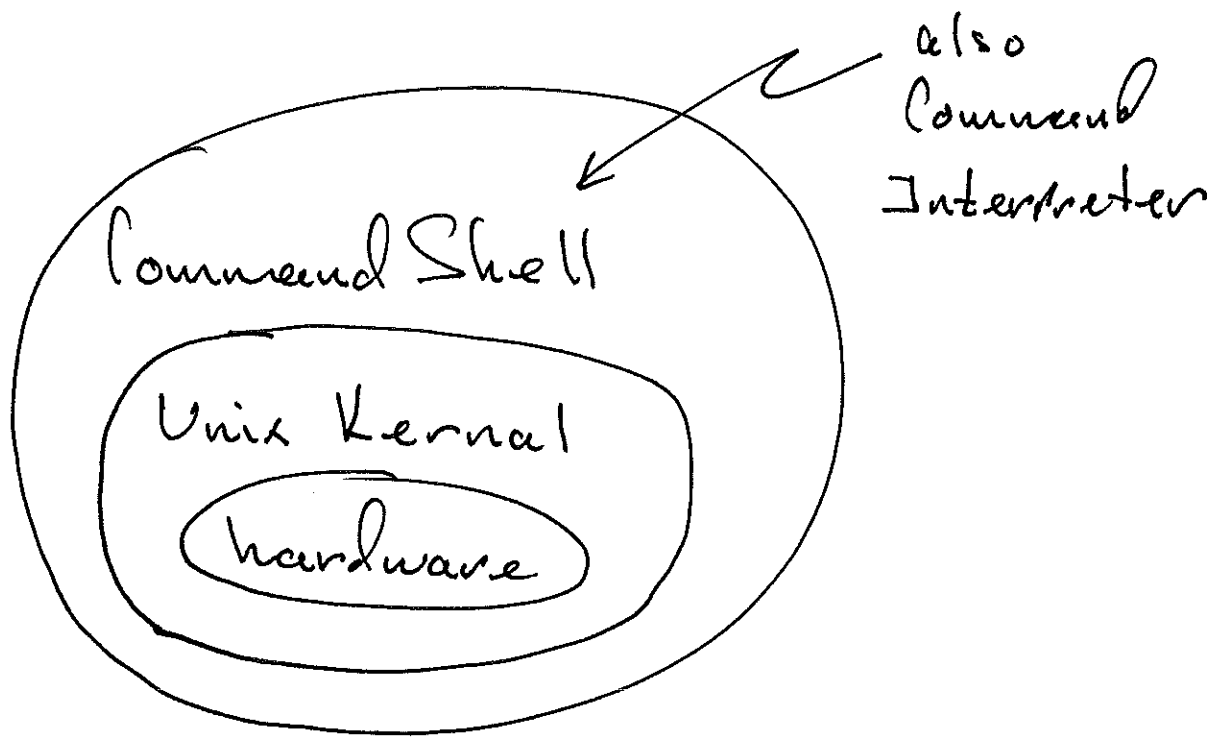
% myProg

or

% a.out

load & Run

Unix Environment :



Command shells:

- csh
- tcsh
- bash } favorites .
- ksh
- sh.

To change your default shell

% chsh shell-name

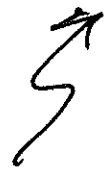
Form of a C++ Program

3

```
//  
// comments  
//  
#include <iostream> ← Preprocessor  
using namespace std;  commands  
  
#define - - - - - ← Define Constant  
                      macros.  
  
classes } not covered  
functions }  
  
int main(void) {  
    // variable declarations  
    :  
    // executable stmts  
  
    return 0;  
}  
classes } not covered  
functions }
```

Constant Macros

```
#define PI 3.14
```



Does literal
text substitution
for "PI"

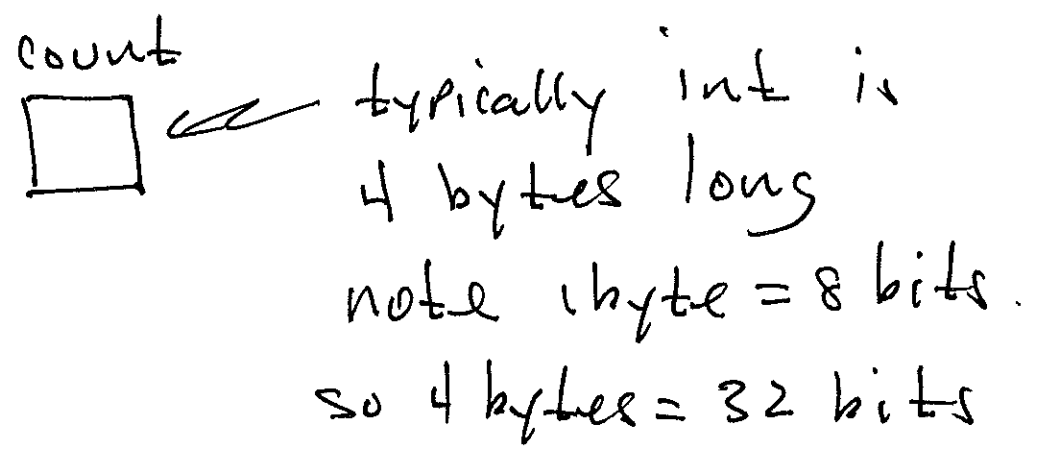
Variable Declarations

Define symbolic names for data areas in memory.

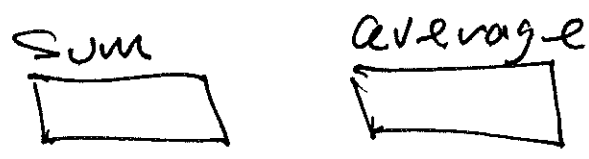
~~Ex.~~

data_type identifier, identifier, ...;

Ex. int count; ← signed integer



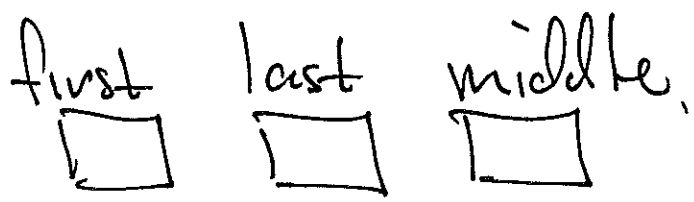
Ex. double sum, average;



Floating Point numbers

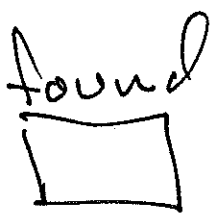
Ex. char first,

last,
middle;



ASCII
character type
1 byte long.

Ex. bool found;



boolean : two values
true
false.

rules for identifiers:

• a legal C++ identifier consists of only:

(1) letters a, ..., z, A, B, ..., Z

(2) digits 0, 1, 2, ..., 9

(3) underscores _

• also: an identifier may not

- begin with a digit

- be a reserved word

Ex. Valid:

happy, Happy, happy1, happy-happy

Ex. Invalid:

1happy, return, int, "happy"

literal values :

int: 1, 2, 57, -379

double: 1.0, 2.0, 57.0, -379.0

char: 'a', 'b', 'c'

bool: true, false

we can initialize at same time as declaration

Ex. int count = 0; count
0

Ex. double sum = 0.0, average;

sumaverage
0.0

Ex. char first = 'p';

19

first
'p'

Ex. bool found = false;

found
false

Symbolic constants

Ex const double pi = 3.14;

Executable Statements

- Input
 - Output
 - Calculation / assignment
 - Conditional
 - Iterative
- } sequential

Input & Output:

input stream: cin
output stream: cout

Ex.

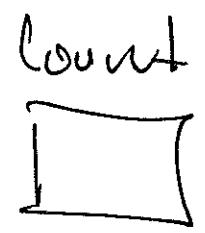
```
int count = 5;  
:  
cout << count;
```



↑
extraction operator

Ex.

```
int count;  
:  
cin >> count;
```



↑
insertion operator.