CMPS 10 11-12-08

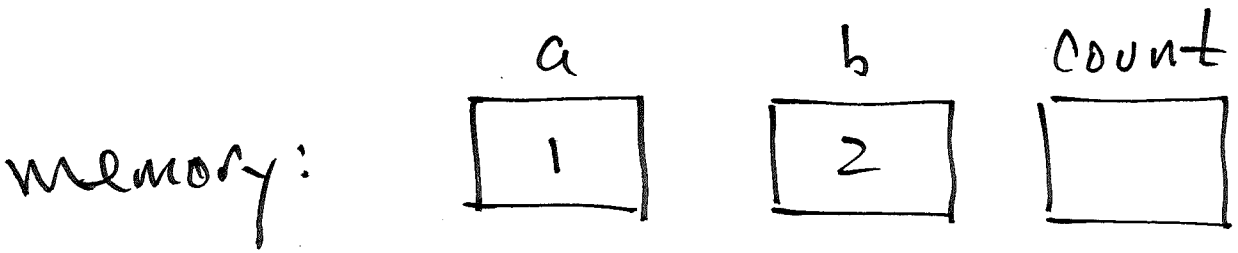
```
#include<iostream>
using namespace std;
:
int main( ) {
    // VARIABLE DECLARATIONS

    // EXECUTABLE STMTS


    return 0;
}
:
```

DATA TYPES:

int	SIGNED INTEGER
double	FLOATING POINT REP. OF REALS
char	CHARACTER CODE
bool	BOOLEAN TRUE/FALSE

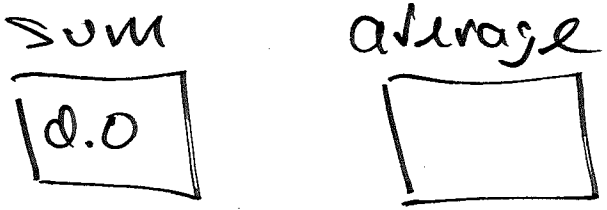
VAR. DECLARATION: data_type var1, var2, ... varn;

Ex. int a = 1, b = 2, count;



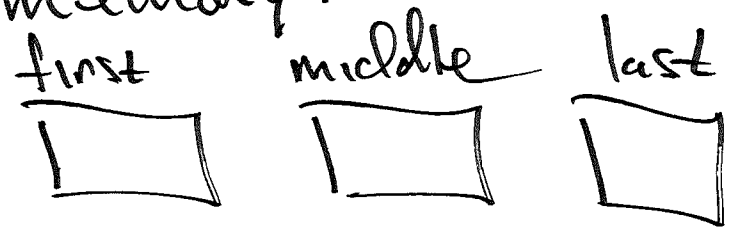
Ex. double sum = 0.0,
average;

memory:

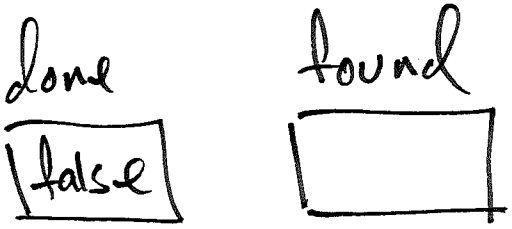


Ex. char first,
middle,
last;

memory:



Ex bool done = false,
found;



Literal values:

int	1, 2, 3, -1, -2, -3, 0 ....
double	1.0, 2.0, ..., -3.0, ... 0.0 ...
char	'a', 'b', 'c', ..., 'A', 'B', ... ')', ':' ...
bool	true, false

Variable names: identifiers.

Rules for valid identifiers:

- must consist of letters, underscores, or digits: a, ..., z, A, ..., Z, _, 0, 1, ..., 9

- MAY NOT BEGIN WITH A DIGIT
- MAY NOT BE A RESERVED WORD

EX (WRONG!!) int int;

SOME VALID IDENTIFIERS:

happy, Happy, happy1, happy_happy
Interest_Rate

some invalid IDENTIFIERS:

1happy, return, int, "happy",
Interest Rate

Symbolic constants: use const

Ex. const double a = 6;

const double pi = 3.14;

a
60

pi
3.14

EXECUTABLE STATEMENTS:

(1) SEQUENTIAL { INPUT, OUTPUT, CALCULATION (ASSIGNMENT)

(2) CONDITIONAL { if, if-else

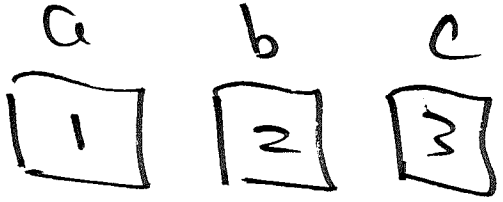
(3) ITERATIVE { while, do-while, for

OUTPUT: Cout

EXTRACTION OPERATION

```
Cout << variable;
Cout << expression;
Cout << literal value;
```

Ex. `int a = 1, b = 2, c = 3;`




```
Cout << a;
Cout << b+c;
```

Ex STRING LITERALS: ENCLOSED IN " "

```
Cout << "Enter temperature: "
```

Input: cin

```
cin >> variable;
```

INSERTION OPERATOR

Calculation / Assignment

Ex.

```
int a=1, b=2, c=3, d=4;

d = a+b-c;
```

ASSIGNMENT OPERATOR

NOTE: COMPARISON OPERATOR ==