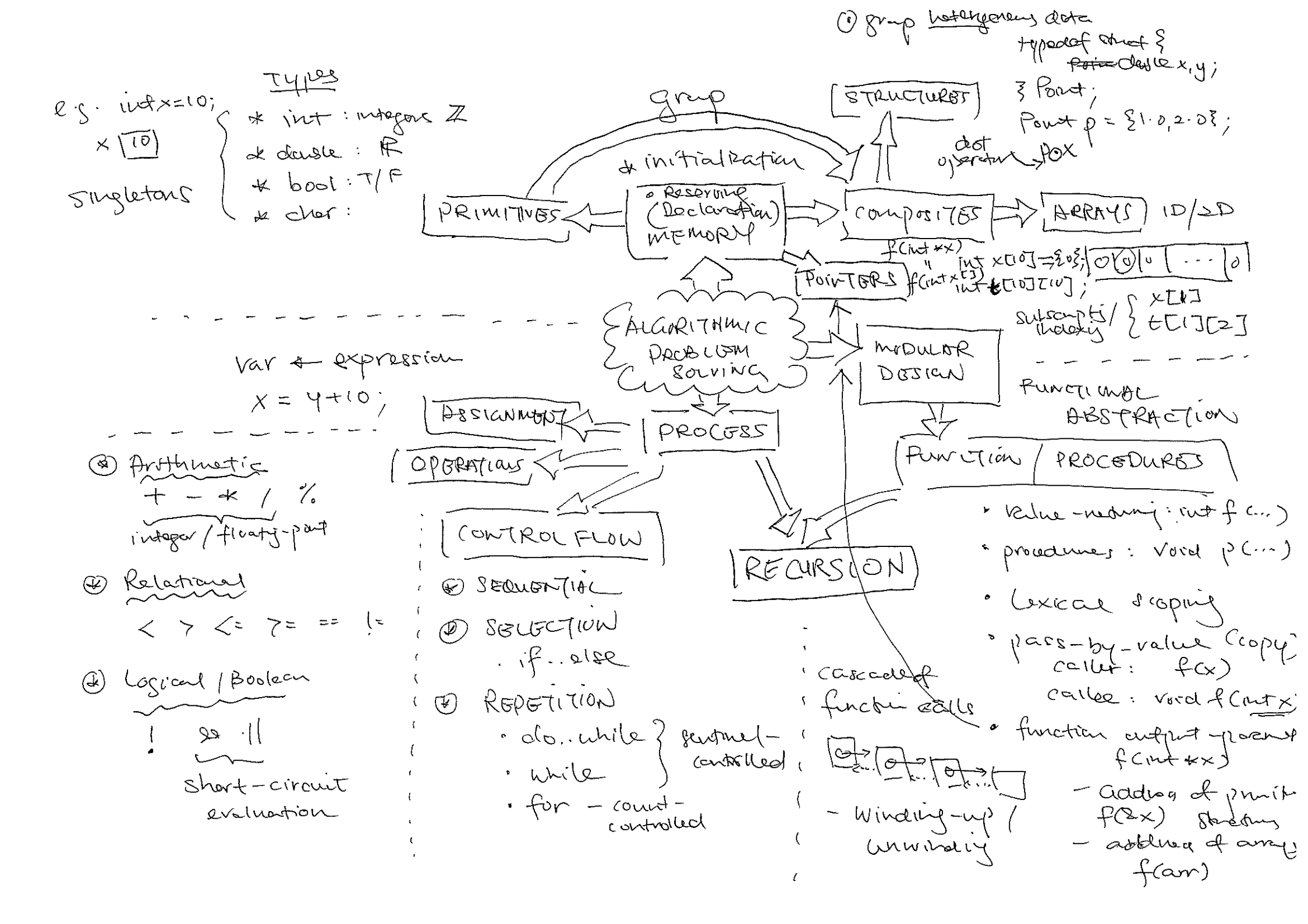
Types
* int : integers ℤ
* double : ℝ
* bool : T/F
* char :
e.g. int x=10;
x 10
singletons
group
STRUCTURES
group heterogeneous data
typedef struct {
double x,y;
} Point;
Point p = {1.0, 2.0};
dot operator → POX
* initialization
PRIMITIVES
• Reserving (Declaration) MEMORY
COMPOSITES
ARRAYS 1D/2D
POINTERS
f(int *x)
f(int x[])
int x[10] = {2,0};
int *t[10][10];
subscript/ indexing
x[k]
t[1][2]
ALGORITHMIC PROBLEM SOLVING
MODULAR DESIGN
FUNCTIONAL ABSTRACTION
Var ← expression
X = Y+10;
DESIGNMENT
PROCESS
OPERATIONS
FUNCTION / PROCEDURES
• value-returning: int f(...)
• procedures: void p(...)
• lexical scoping
• pass-by-value (copy)
caller: f(x)
callee: void f(int x)
• function output-process
f(int *x)
- address of primitive f(&x) sharing
- address of array f(arr)
① Arithmetic
+ - * / %
integer / floating-point
② Relational
< > <= >= == !=
③ Logical / Boolean
! && ||
short-circuit evaluation
CONTROL FLOW
① SEQUENTIAL
② SELECTION
• if..else
③ REPETITION
• do..while
• while
• for – count-controlled
sentinel-controlled
RECURSION
cascade of functn calls
- winding-up / unwinding