

Symbol Table

1. Variable
2. Function
3. Data type
4. Object
5. Class
6. Interface

Symbol Table implementation

1. List (sorted & unsorted)
2. Hash Table
3. Binary Search Tree


→ Sorted



Symbol Table

Linked List

```
1 main ()
{
2 char a [5 ] ;
3 int x
4 x = 5
5 print (x)
```

Name	Type	Size	Dimension	Line of Declaration	Line of Usage	Address
a	char	5	1	2	—	memory location
x	int	4	0	3		memory location

Symbol Table Operations

1. Insert ()
2. lookup ()
3. Delete ()
4. Scope Management

① insert (a, int)

② lookup (a) → Search

No Duplicate

SDD to Store Information Type

$D \rightarrow D_1, id \quad \{ \text{Add type (id, } D_1 \text{ type)}, D \text{ type} = D_1 \text{ type} \}$

$D \rightarrow T, id \quad \{ \text{Add type (id, } T \text{ type)}, D \text{ type} = T \text{ type} \}$

$T \rightarrow int \quad \{ T \text{ type} = int \}$

$T \rightarrow char \quad \{ T \text{ type} = char \}$

$T \rightarrow float \quad \{ T \text{ type} = float \}$

input : int x, y, z

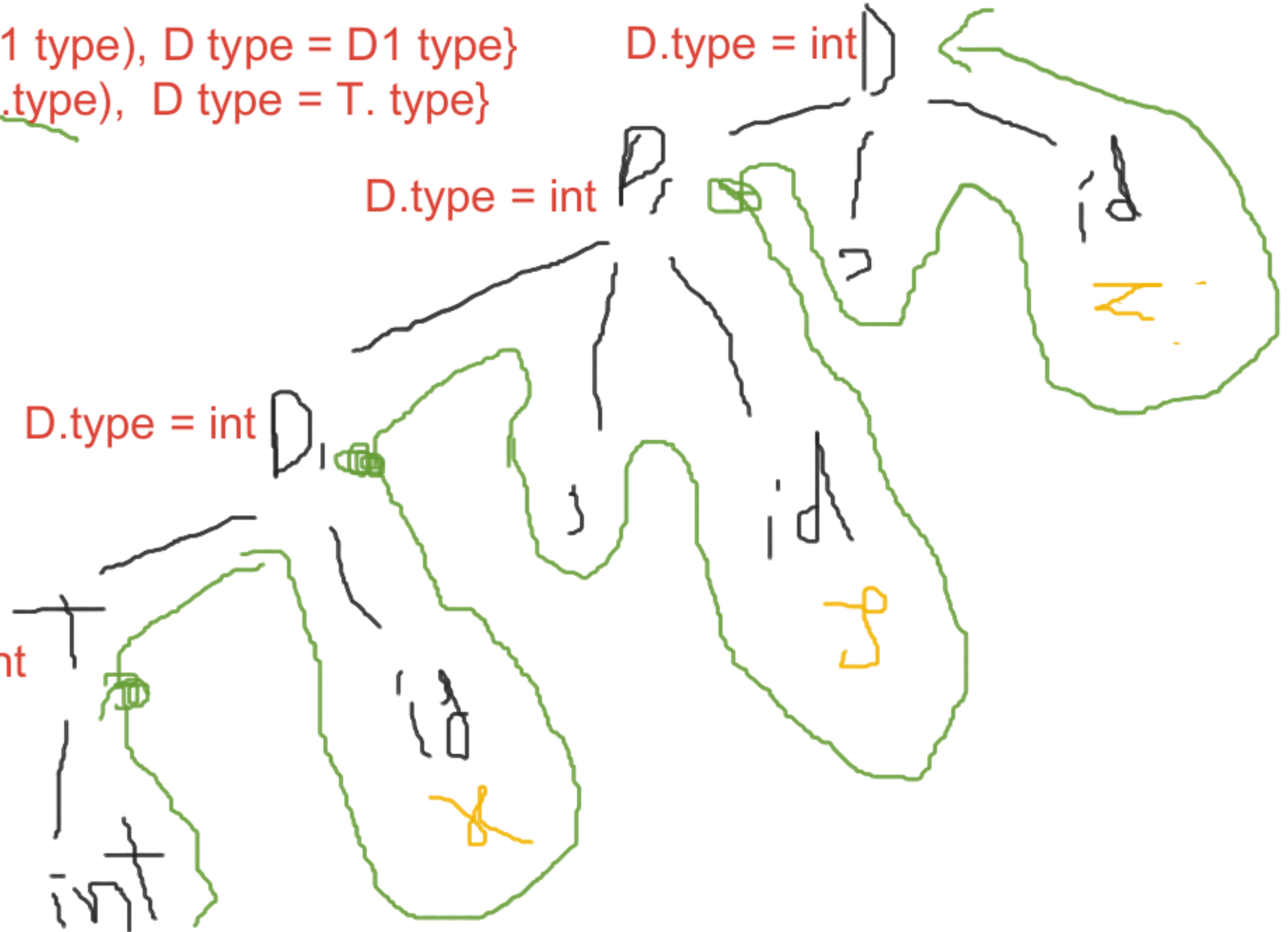
v. name	Data type
x	int
y	int
z	int

T.type = int

D.type = int

D.type = int

D.type = int



Scope management

```
int value;
void one ( )
{
    int a;
    {
        int b;
    }
    int c;
    {
        int d;
    }
}
Void two ( )
{
    int e;
}
```

Global Symbols		
Name	Type	Data Type
Value	variable	int
one	Function	void
two	Function	void

Function One		
name	Type	Data type
d	variable	int
c	variable	int

Function Two		
name	Type	Data type
e	variabl	int

b	variable	int
---	----------	-----

d	variable	int
---	----------	-----

Scope management

```
int x
int y
void sum (int m){
    float x,y
    if(-----){
        int i,j
    }
    else {
        double i,j
    }
}

int g (int n){
    bool flag;
}
```

Global Symbols		
Name	Type	Data Type
x	variable	int
y	variable	int
sum	function	void
g	function	int

function
sum

function g

Name	Type	Data Type
m	arg	int
x	variable	float
y	variable	float

Name	Type	Data Type
n	arg	int
flag	variable	boolean

Name	Type	Data Type
i	variable	int
j	variable	int

Name	Type	Data Type
i	variable	double
j	variable	double

