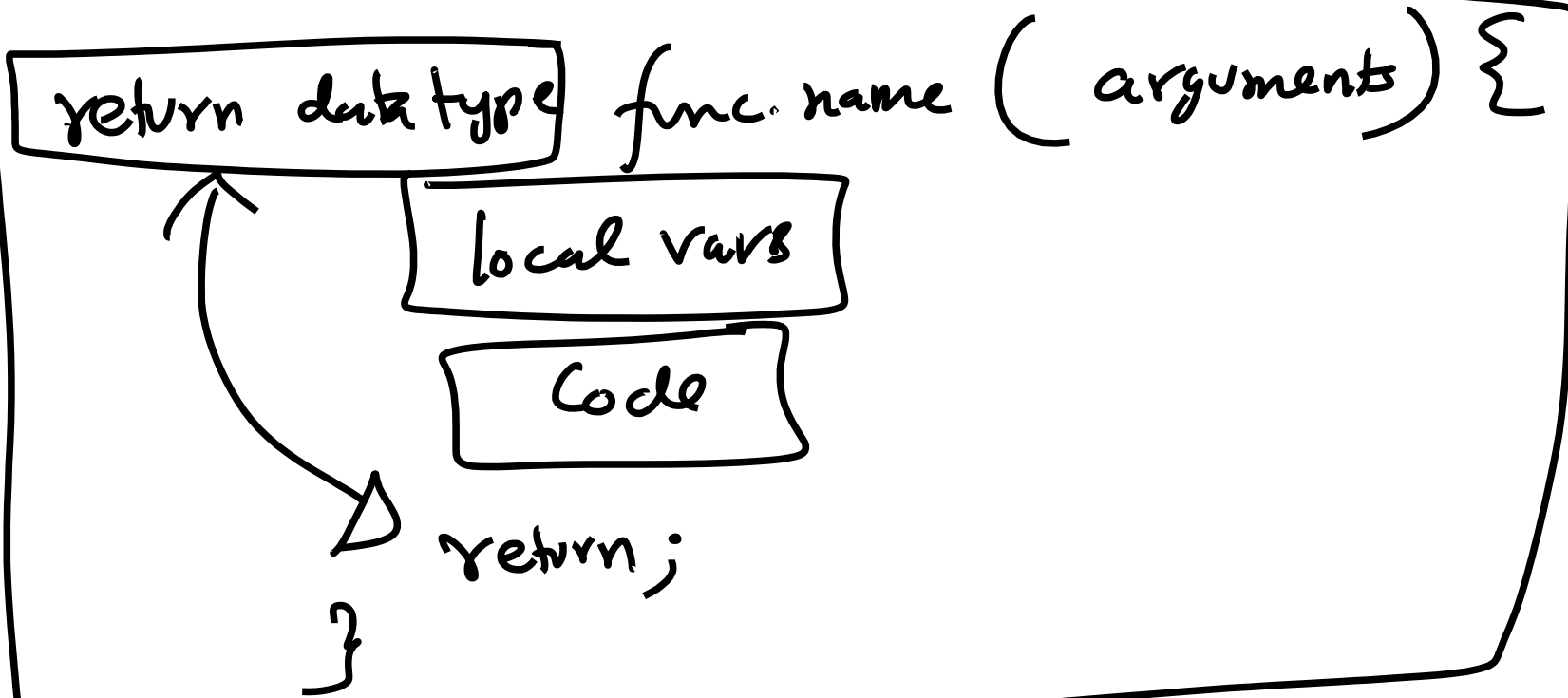


02/Nov/2017



① passing by Value

main() {

int x = 10;

==
printf(x); → x = 10

func(x);

printf(x); → x = 10

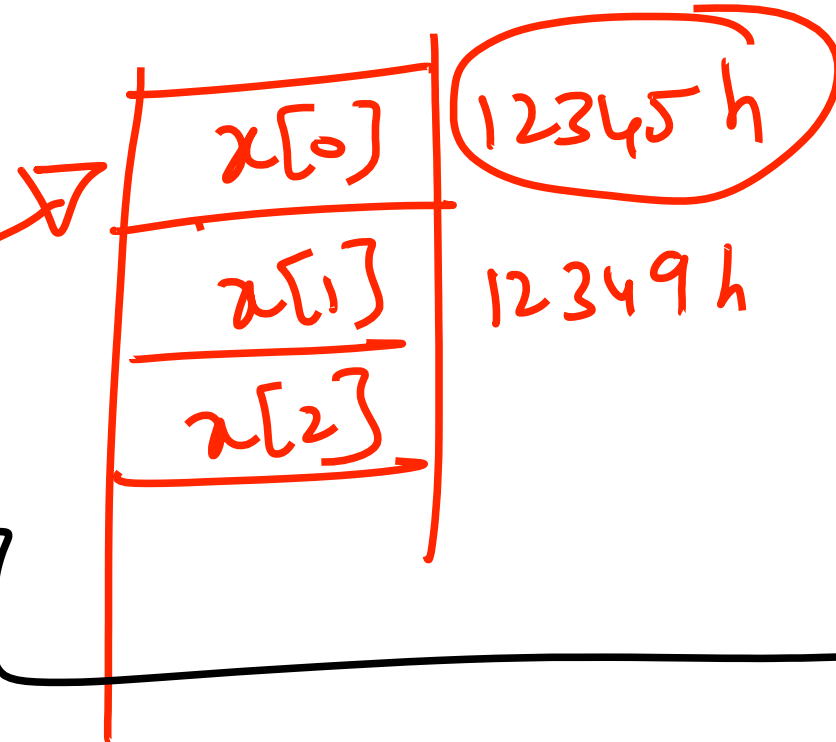
==
}

void func(y) {
==
==
==
y = y + 1;
printf(y);
return;
}

② array passing

int x[3] = {1, 2, 3};

func(x);



`x * y;`
`(5+2)*(5+2)`

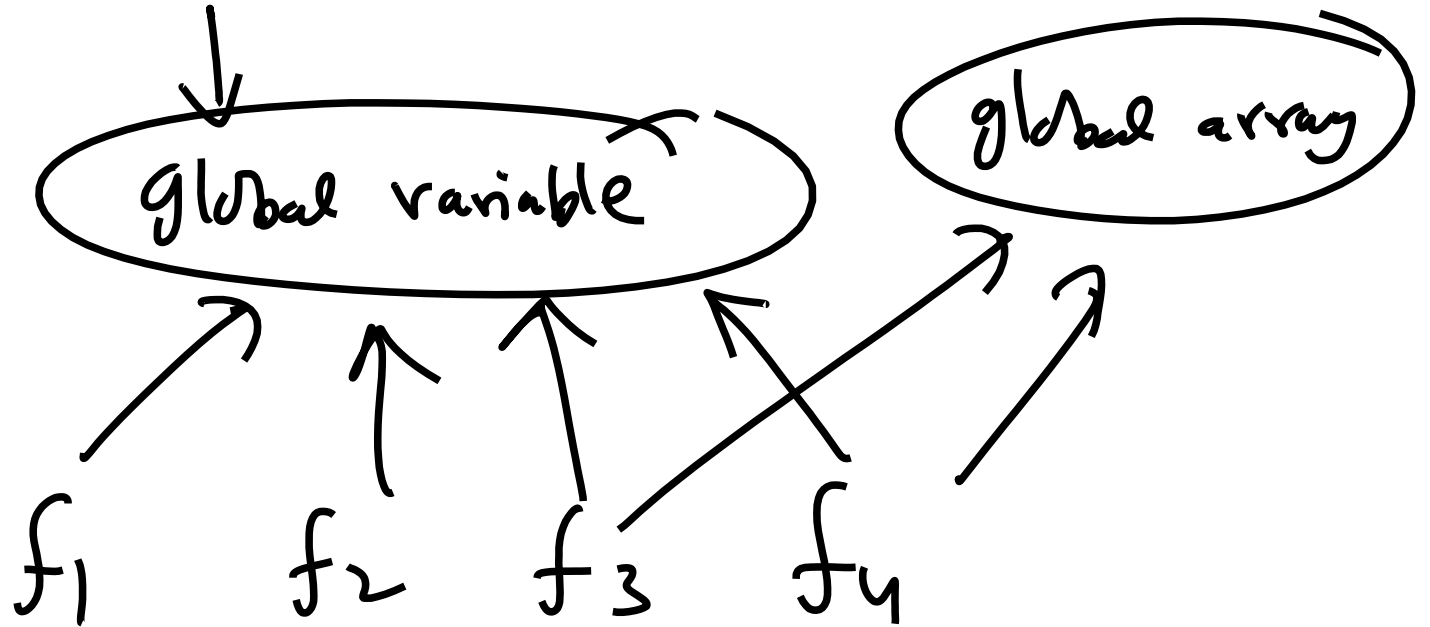
```
#define inc(x) (x++)  
:  
main() {  
    ==  
    (x++) inc(x);  
    ==  
    (x++) inc(x);  
    ==  
    ...  
}
```

Diagram illustrating the macro `inc(x)` and its usage in `main()`. The macro is defined as `#define inc(x) (x++)`. In the `main()` function, the macro is used in two ways: `(x++) inc(x);` and `(x++) inc(x);`. Arrows indicate that the `(x++)` part of the macro call is expanded to `x++` in the code.

Scope of a variable

```
func ( ) {  
    int x;  
}
```

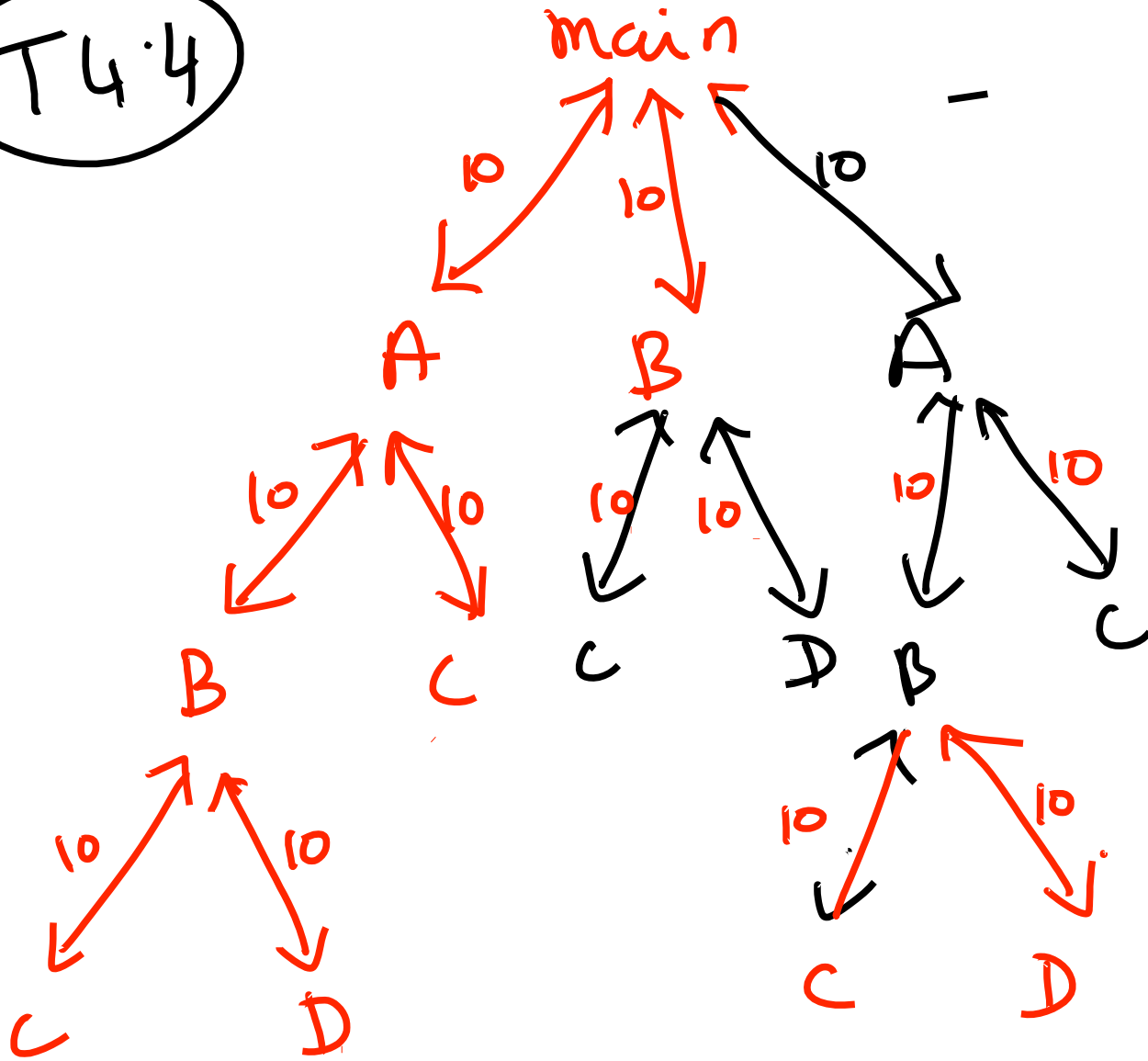
"Extern" → External



```
{ float arr[10];  
  int x;  
  main ( ) {  
      ≡  
      }  
}
```

← global vars.

T4.4



$$\left\{ \begin{array}{l} 1200 \text{ — } \text{main}() \\ 13 \times 80 = 1040 \\ 13 \times 10 = 130 \rightarrow (\text{call-ret}) \end{array} \right.$$

Total time =
 $(1200 + 1040 + 130) = 2370 \text{ msec}$

Overhead = $130 / 2370 = 0.0548$
 $\sim 5.5\%$