Note Title 20-08-2018

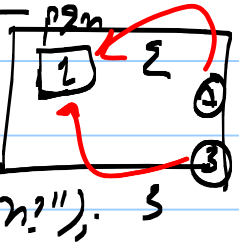
$2^\wedge 0 = 1$ 1 | i=0
$2^\wedge 1 = 2$ 1 | i=1
$2^\wedge 2 = 4$ 1 | i=2
$2^\wedge 3 = 8$ 1 | i=3
$2^n = 16$ 1

COL100
Aug 20, 2018

"Scope in c++ is static."

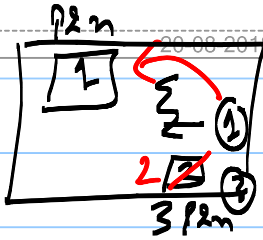
outside the loop

Last class:- "Scope"



```
int main() {
→ int p2n= 1;
  int n= getInteger("n?");
  for( int i=0; i<n; i++) {
     → int tmp = p2n;
       cout<< "2^" << i << "=" << tmp
       << endl;
       p2n = tmp * 2;
  }
  cout<< "2^n =" << p2n << endl;
}
```

n=4



```
int main() {
    int p2n= 1;
    int n= getInteger("n?");
    for(int i=0; i<n; i++){
① →   int tmp= p2n;
②     cout<< "2^n" << i
          << "=" << tmp << endl;
③     int p2n = tmp*2;
    }
    cout<< "2^n=" << p2n
          << endl;
}
```

Arguments

2

int

One more example:-

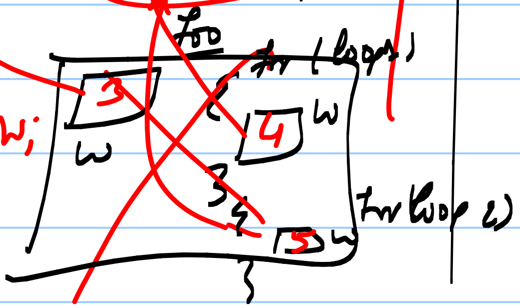
```
void foo( ~~) {~~ int w) {
    ~~int w = 3;~~
    for (int i = 0; i<5; i++){
        int w=4;
    }
    for (int i=0; i<2; i++){
        int w=5;
    }
}  -> return w;
```

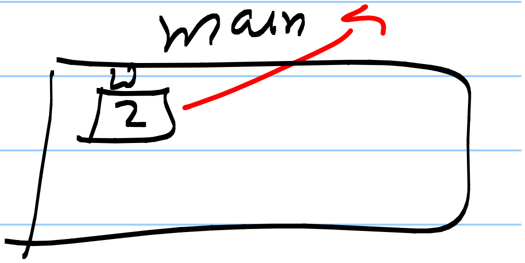
return type

```
int main () {
    -> int w=2;
    -> foo(w);  -> "calling"
    -> cout<<w<<endl;
}
```

Return type

2 is printed





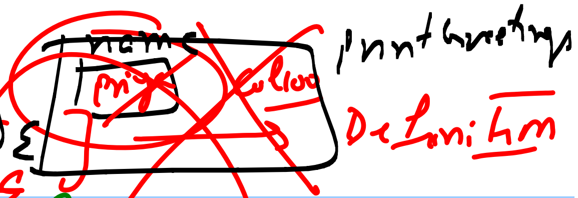
function

type

Declaration

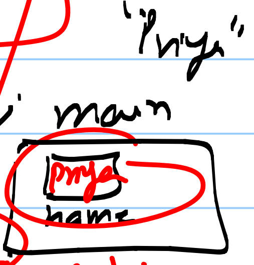
void printGreetings(string name){ (int n) { a
→ ~~string name~~
cout << "Hello World" ~~<<endl;~~ << name << endl;
cout << "Have a great day" << endl;
name = "col 100";
}

"parameters are copies"



int main() {
→ int n = getInteger("n");
→ string name = getLine("name?");
while (name != "") {
cout << name << endl;
→ printGreetings(name, n);
→ cout << name << endl;
→ name = getLine("name?");
}
Cout << name << endl;
}

caller

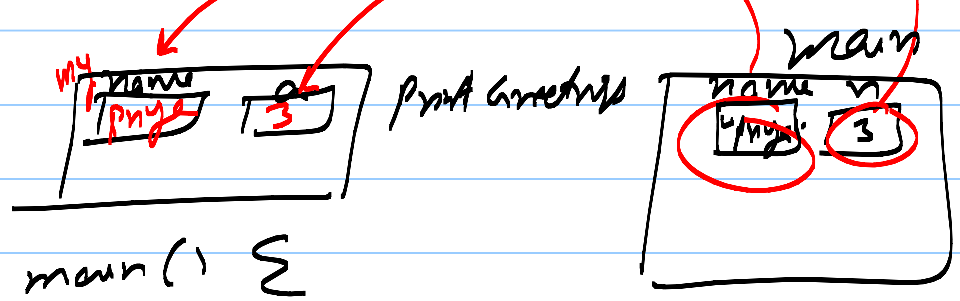


```
void printGreetings(string myname, int a) { [ERROR!
        cout << name << endl;

}

int main() {

    int n = getInteger("n:");
    string name = getLine("name:");
    ~~for~~ while(name != "") {
            printGreetings(name), ~~int~~, n, 3);
    }
```



General template

return-type

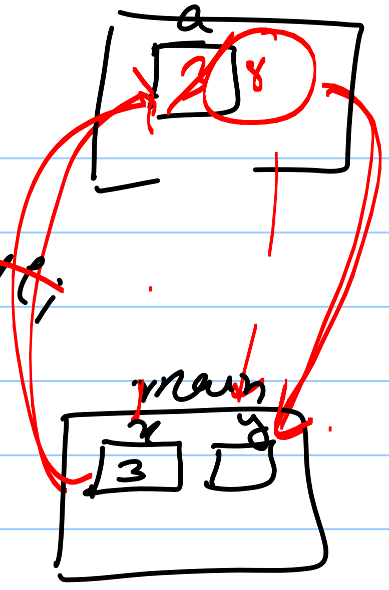
~~void~~ function_name(type1 arg1, type2 arg2,
. . . typen argn)

{

}

```
int
void  addFive(int a) {
→   a = a + 5;
    cout << a << ~~endl~~;
    return a;
}

int main() {
    int x = 3;
→ ~~int y~~  ~~x =~~ addFive(x);        ~~3+5~~;
    cout << x << endl.      // 3
    cout << y << endl;      // 8
          int b = 5 + 3;
}
```


a
2
8
main
x
3
y