COL 100
Aug 16, 2018

col100-staff @ iitd.ac.in

Last Class:-
Flowcharts
Loops (Nested)

method/
function

Functions / Methods

return type

arguments

```
(int) printGreetings() {          function declaration/
                                   definition
    cout << "Hello World" <<endl;     function body
    cout << "Have a great day" <<endl;
    return 0;
}

int main() {
    string name = readline("Next?");
    while ( name != " ") {
        printGreetings();             get
        name = readLine("Next?)
    }
}
```

call
? ?

```cpp
int main () {

    string name = getLine("Next Name?");
    while ( name != " ") {
        cout << name << endl;
        cout << "Hello, World" << endl;
        cout << "Have a great day" << endl;
        name = getLine("Next Name?");
    }
    return 0;
}
```
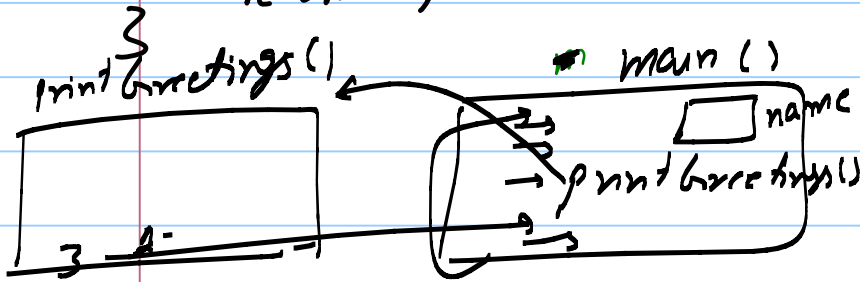
→ printGreeting();

printGreetings()

3

main ()

□ name

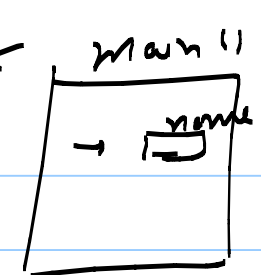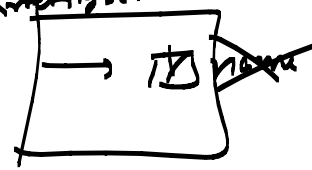printGreetings()

3   1

$> Next Name? Calloo
$> COf 100
$> Hello, World
$> How a great day
$> Next Name? L
$>

```cpp
void   printGreetings () {
    cout << "name is!"  << name;
    cout << "Hello, World" << endl;
    cout << "Have  a great day << endl;
}
```

name is not defined!

```cpp
int   main () {
    string  name = getline(", next?");
    while( name != "") {
        printGreetings();
        name = getline(" Next?");
    }
}
```

scope is limited to the main function

Scope of a variable:

Extent to which a variable is "Valid" in the program

→ Scope **starts** when the variable is defined

→ Scope ends at the termination of block in which it is defined
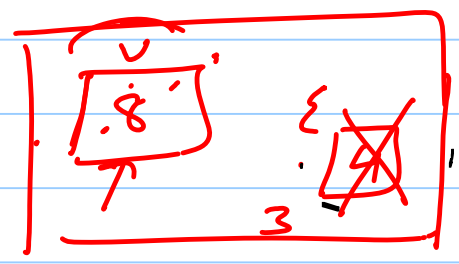
→ block :- chunk of code enclosed in { } code

```
int main() {
                    V = 8;
    double  V;                ]  → shadowing
                                    a variable
    if ( condition ) {
        → V = 4;
            -- some code
    }
    V = 1;
          some code
    cout << " Outside value
          of v=" << v;
}
```

start "scope of v"

Then "scope of v"
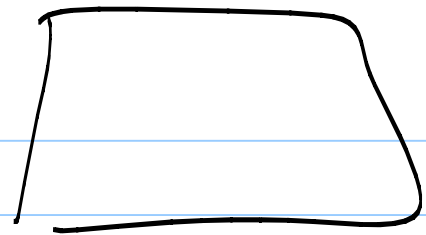
```
int i = 0;
if( true) {
    →        i = 1;
"bind"  →  cout << i <<endl;   // 1

    if ( true) {
    →        i = 3;             //
        cout << i <<endl;      // 3
    }
    3
    cout << i <<endl;    → // 1  // 3
3  cout << i <<endl;             // 3
```

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

3

3

```cpp
int main() {
    int w = 2;
    foo();
    3   cout << w;   // 2
}
```
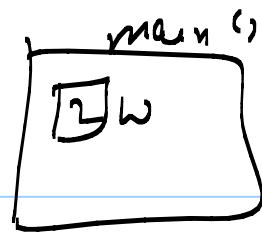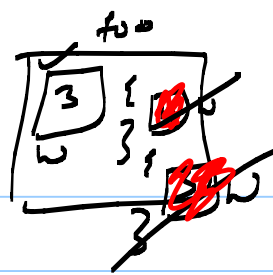
```cpp
→ int i;
→ for (    i=0;   i < 5 ;  i  (i++)  {

      →  cout << i;


   →  }
      → cout << i
        int main ()
        {
   →  double u=5 ;
   →  cout << u;                Error
   →  double u=7
        cout << u;'
   }
```

```
for (int i=0; i<5; i++) {
    cout << i << endl;
    for (int i=0; i<2; i++) {
        cout << i;
    }
    cout << endl;
}
```

3

cout<<endl:
```
0
0  1
1
0  1
2
0  1
3
4  1
```