COL100 Lecture 9

Note Title 20-08-2018

Review:

Scope / Lifetime

{

code block

declaration != assignment

}

Variable shadowing

Parameters to Methods

```
int x = 5
for ( int i = 0; i < 10; i++)
{
                                        int x;
                                        x = x + i;  | Local
                                                    | to this
                                                    | code
                                                    | block
}
```

```
void foo()
  {
        int x;

  }

void bar()
  {
      int y;

  }
```

```
{
    int x;   → global variables
    void foo()
     {

     }
     void bar(){
      {

      }
}
```

```
string  name;
void   printGreeting ()
{
        cout <<   "Hello " << name << endl;
}
int  main()
{
      while (true)
      {
          name = getLine ("you name?");
          printGreeting () ;
      }
      return 0;
}
```

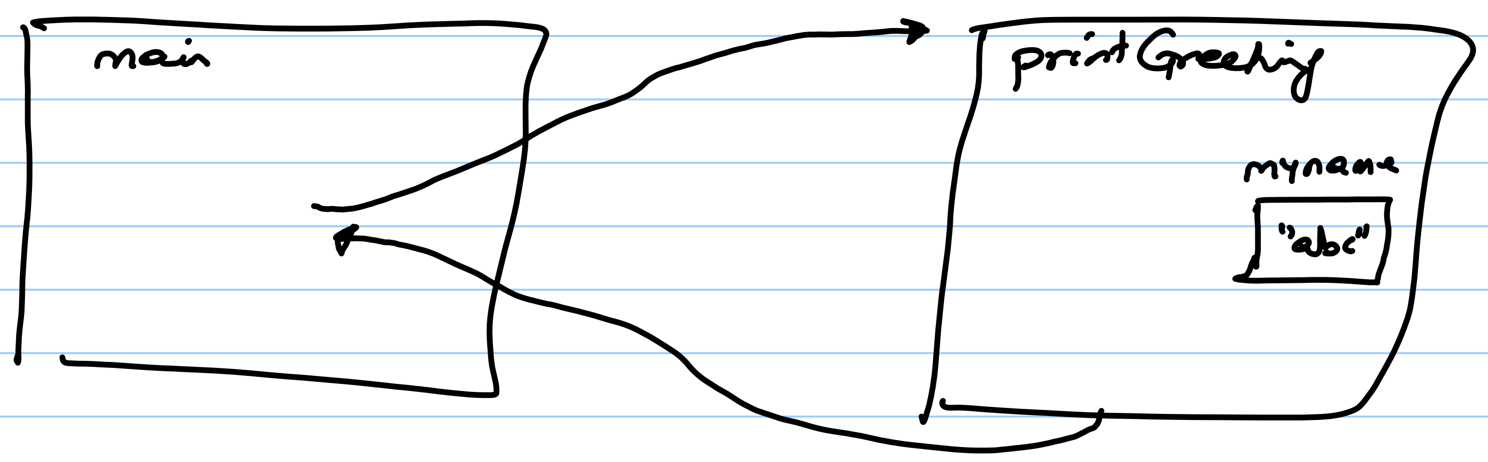
Parameters

DECLARATION

```
void   printGreeting (string  myname)      → variable
{
          cout  <<  "Hello  "  <<  myname << endl;
}
```

FUNCTION CALL

```
int   main()                                 → value
{
          printGreeting ("abc"); // function call
          return 0;
}
```


main
printGreeting
myname
"abc"

string x = 1.0; X

```
void printGreeting(string myname, ~~string~~ int mycourse)
{
    cout << "Welcome " << myname
         << " to " << mycourse
         << endl;
}
int main()
{
    printGreeting(1.0, false); X
    printGreeting("abc", "col 100");
    return 0;
}
```

string myname; X

string myname;

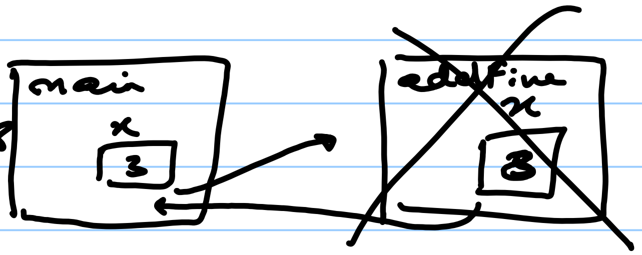
parameters / arguments

```
ret_type   name ( type1 arg1 , type2 arg2, ... , type_n
                                                 arg_n)
{
        statements;

}
name ( val1 , val2, ... , val_n ) ;
```

```
void   printGreeting ( string myname , int  times)
{
        for (int   i = 0;  i < times;  i++)
         {
                cout  <<   "Hello  "<<  myname ;
                                          <<endl;
          }
 }

int    main()
{
        printGreeting ("abc",   5);
         return  0;
 }
```

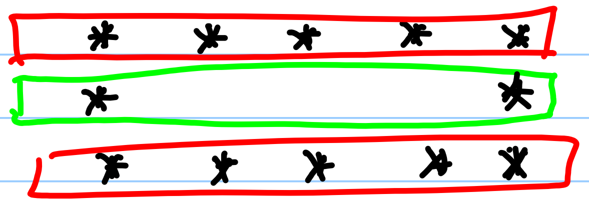
Parameters are copies!

```
void    addFive (int    x/y)
{
            x/y = x/y + 5;
            cout << x << endl; //8
}
int    main()
{
          int  x = 3;
           addFive (x);
           cout  <<  x/y << endl;          // 3
           return 0;
}
```

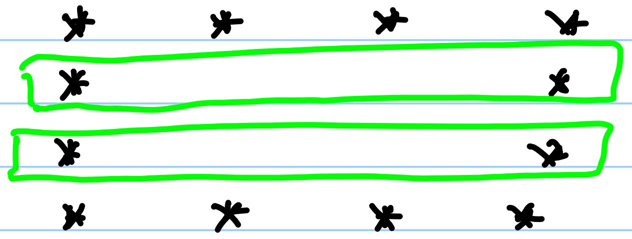


__drawbox__ (int height, int width)

drawbox (3, 5)



drawbox (4, 4)



drawLine

draw Edge

```
void  drawbox (int  height ,  int  width)
{
          drawLine ( width) ;
          for (int  i = 0;  i< height -2 ; i++)
           {
                drawEdge( width);
           }
          drawLine ( width);

  }
```

```
void    drawLine (int  width)
{


  }

 void    drawEdge (int   width)
{


 }
```

Return Values

```
double  metersTocm ( double  meters)
{
        return  100 * meters;

}

int  main()
{
        double  meters =  getDouble(" meters?");
        cout  <<  metersTocm( meters) <<  " cm. "<<endl;
        return 0;
}
```

expression
just 2+3

```
~~void~~
~~do~~  ~~int~~
double    metersTocm  (double  meters)
{         double   cm;
          if    (meters  <100)
          {        cm =  100*  meters;
          } else
          {
                 cout <<"invalid  meters value" <<endl;
~~}~~                cm = 0;
           }
          return  cm;
 }
```

Parameters vs. Return values

Parameters: send information "in" to a method
from the caller

Return values: send information "out" of a
a method to
the caller

```
string  getLine ( . . . )
int     getInteger ( . . . )
double  getDouble ( . . . )
```

abs : absolute value

abs(0) = 0
abs(10) = 10
abs(-20) = 20

```
~~int   abs (int   x)
{     if  (x <0)

 }~~
```

```
int    abs ( int    x)
{
       int    ret,
       if   (x   >=0)
       {
              ret = x;
       } else {
             ret = -x;
       }
       return  ret;
}
```

```
if   (x >= 0)
{
   return x;
} else {
   return  -x;
}
```

```
int     round ( double   x)
{


        }
```

```
bool checkEven( int x)
{


 }
```

Return statements can occur in the middle of the code too! | %

```
bool      checkEven (int   x)
{
          if     ( x % 2 == 0 )
          {
                 return true;
          } else {
                 return false;
          }
          cout << "Hello" << endl;   return false;
}
```

Unreachable code

% : mod operator

7 % 2 == 1
6 % 2 == 0

7 % 3 == 1

```
int   addFive (int  x)
{
      return   x+5;
 }

 int   main()
 {
         int   x=3;
   x =   addFive(x);
          x = 2 * x;
          cout  << x ;      // 16
  }
```

Methods
Parameters
Return Values
Declarations
Function Calls

Declaration order

```
int   main()
{
      printGreeting(); //X: undefined method
}
void printGreeting()
{ ...
}
```

__Function prototypes__ . includes everything from a declaration up to the first curly {

```
void    printGreeting (string myname, int times);

int     main()
{
            printGreeting ( "abc" , 5 ); // ✓
            return 0;
}
void    printGreeting ( string myname, int times)
{
        : : :
}
```