CS 31 Discussion 1J

ABDULLAH-AL-ZUBAER IMRAN & RYAN YANG

WEEK 3: STRINGS AND FUNCTIONS

Recap

- Strings
  - Concatenation, empty string, cin and cout stream, getline(cin, s), cin.ignore(n, pattern)
- Conditional Statements
  - If…else, if…elseif…else, switch
  - If..else ladder to switch conversion
- Loops
  - while, do…while, for
  - Conversion from one loop to another

Discussion Objectives

Review and practice things covered during lectures

- Char and More about Strings
- Functions
- Coding examples

Programming Challenge

Time for you to ask questions!

Character Datatype

Char

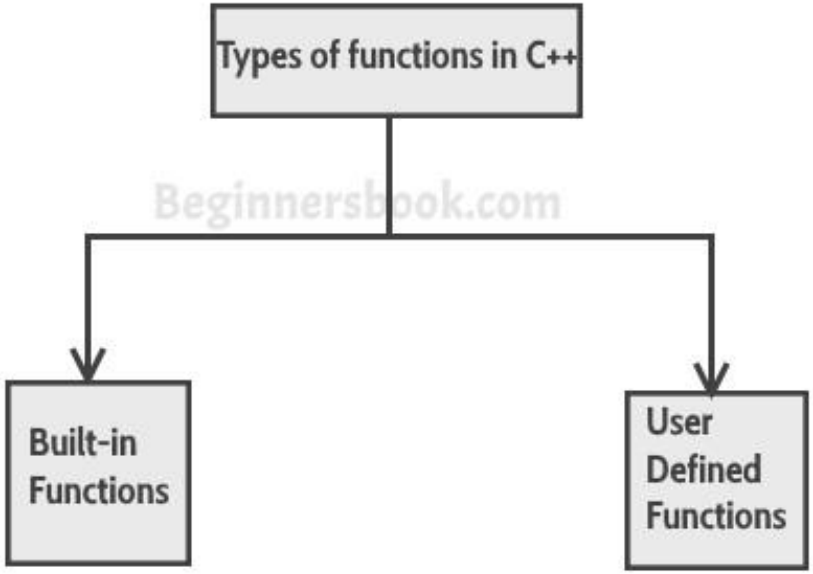
- Takes one byte of memory
- Single letter or digit:
- 'A’ '@’ ’ ‘ '4’ '\n'

```
string s = "Hello";
char c = s[1];  // 'e’
```

More on Strings

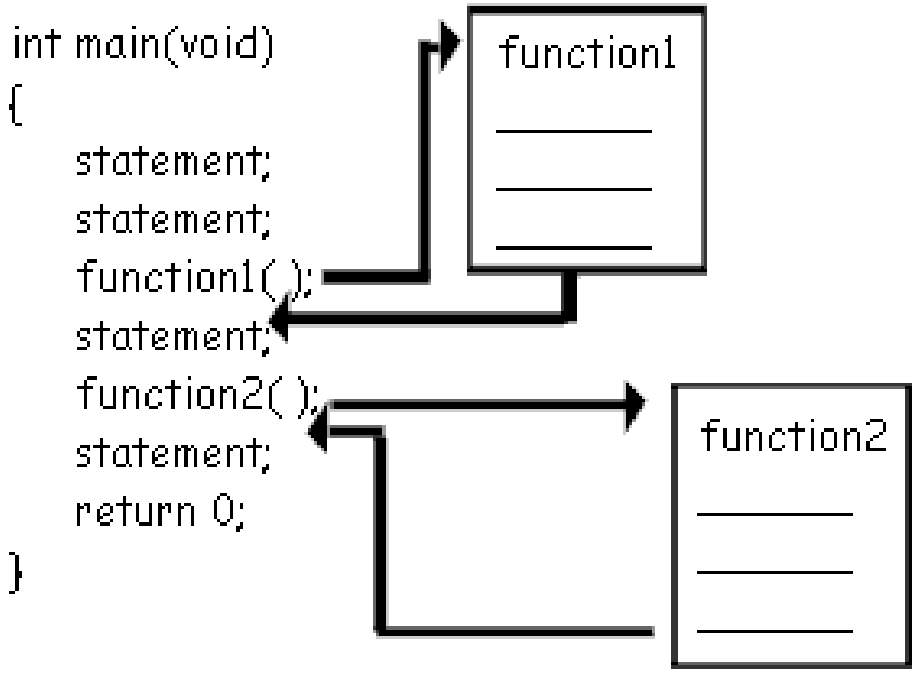
```
#include <iostream>
using namespace std;

int main(){

string s = "Hello";

for (int k = 0; k != s.size(); k++)
        cout << s[k] << endl;

cout << "Enter some text: ";
string t;
getline(cin, t);

int numberOfEs = 0;

for (int k = 0; k != t.size(); k++)
 {
       if (t[k] == 'E'  ||  t[k] == 'e')
       numberOfEs++;
         }

cout << "The number of Es (upper and lower case) is "
<< numberOfEs << endl;

return 0;
}
```

Functions



Predefined and precompiled

Declare
Define
Call



Functions (Cont'd)

```
#include <iostream>

// function declaration
return_type func_name (param_1_type param_1_name,
param_2_type param_2_name, ...);

int main() {
    // function call
    return_type var_x = func_name (arg_1, arg_2, ...);
}

/*
 * Note: The top level comment above a function
 * goes here using the multi-line comment, and usually
 * should describe the function's input and output.
 */
return_type func_name (param_1_type param_1_name,
param_2_type param_2_name, ...) {
    // func_name do stuff
}
```

Functions (Cont'd)

```
// function prototype for foo
int foo(int x);

int main() {
    cout << foo(2) << endl;
    cout << foo(0) << endl;
}

// function implementation for foo
int foo(int x) {
    x *= 2;
    if (x < 100)
        return foo(x);
    return x;
}
```

Functions (Cont'd)

//What would be the purpose of this function?

```
void greet(int nTimes, string msg)
        {
                for (int k = 1; k <= nTimes; k++)
                        cout << msg << endl;
        }
```

```
#include <iostream>
using namespace std;

void foo(int x);
int x = 6;

int main() {
    foo(x);
    int x = 5;
    foo(x);

    if (x > 5) {
        int x = 4;
        foo(x);
    } else {
        int x = 3;
        foo(x);
    }
}

void foo(int x) {
    cout << "x = " << x << endl;
}
```

Global scope

func main scope

if-block scope

else-block scope

func foo scope

Built-in Functions

sqrt	Square root	double	double	sqrt(4.0)	2.0	cmath
pow	Powers	double	double	pow(2.0,3.0)	8.0	cmath
abs	Absolute value for int	int	int	abs(-7) abs(7)	7 7	cstdlib
labs	Absolute value for long	long	long	labs(-70000) labs(70000)	70000 70000	cstdlib
fabs	Absolute value for double	double	double	fabs(-7.5) fabs(7.5)	7.5 7.5	cmath
ceil	Ceiling (round up)	double	double	ceil(3.2) ceil(3.9)	4.0 4.0	cmath
floor	Floor (round down)	double	double	floor(3.2) floor(3.9)	3.0 3.0	cmath
exit	End program	int	void	exit(1);	None	cstdlib
rand	Random number	None	int	rand( )	Varies	cstdlib
srand	Set seed for rand	unsigned int	void	srand(42);	None	cstdlib

Functions FAQ

Where do we define functions?

There are two conventional ways, which are equivalent. The requirement is that the function must be defined before it can be used, just like variables.

So you either **completely define it before the function** is used, or add the **prototype** and define it later in the program.

The prototype is a way of telling your compiler that there is such a function, but that we will define it later. Remember to add a semicolon after the prototype, but not after the function header.

Functions FAQ

I defined the function, why doesn't it run?

Defining a function does not imply using it.

You must explicitly call (or invoke) the function somewhere to see it running.

When you call it, it will be run as you defined it.

Where you call it and how you call it depend on you.

Functions FAQ

Why does the return value not show up on the screen?

Because you did not display it, and it's not meant to be displayed.

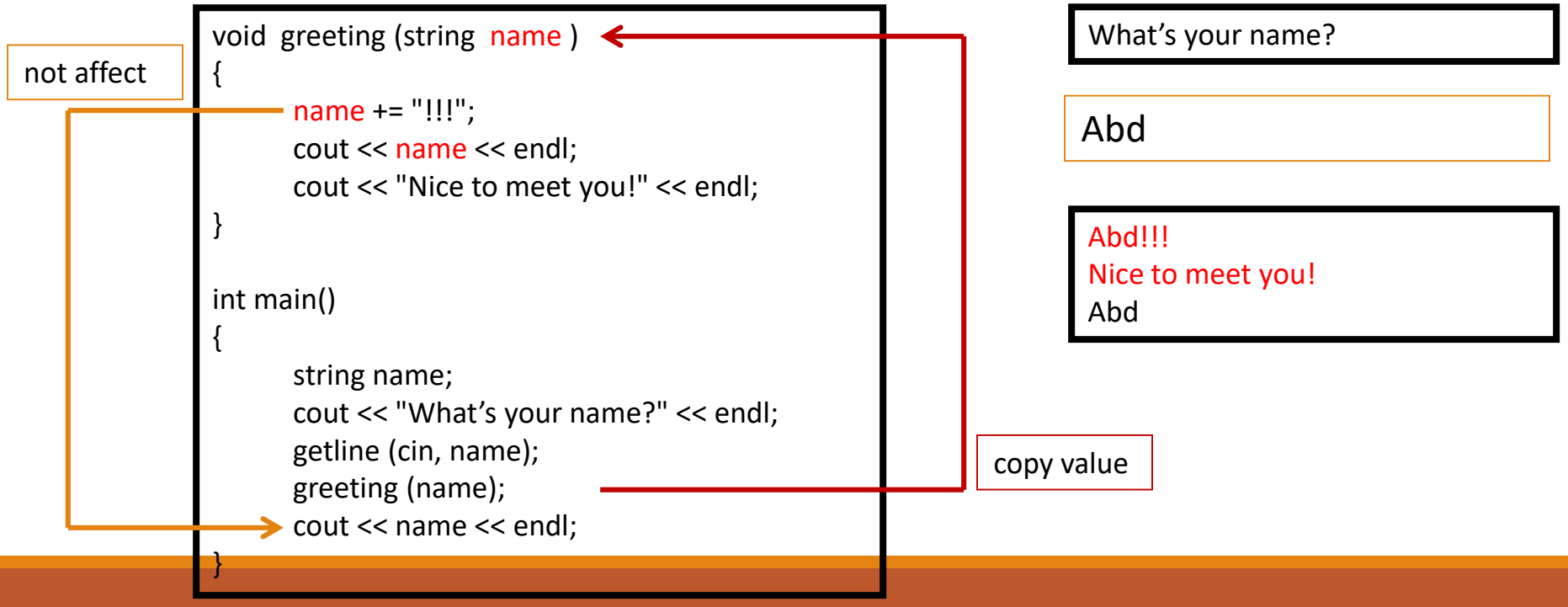
There are people who confusing "returning" with "outputting," which is different.

When you return a value from a function, you return it to whoever called the function.

Passing arguments by value

passing by values into functions

- Doesn't not allow you to access/modify variables outside
- values of arguments exist only in functions



Thanks!

Questions?

Some of the materials presented have been taken from previous TA discussions