

Lecture 16: Calling Functions

Dr. Mohammed Hawa
Electrical Engineering Department
University of Jordan

EE529: Simulating Communication Networks.

Functions

- To write practical large programs (that solve real-world problems), we divide the problem into smaller parts (*functions* or *subroutines*)
 - Predefined functions (called *library*): Both C and C++ standards provide their own
 - User-defined functions
- You need to know how to define a function, call a function, pass parameters to the function, and return results from the function.



Part of C Library

Function	Header File	Purpose	Parameter(s) Type	Result
abs (x)	<cmath>	Returns the absolute value of its argument: <code>abs (-7) = 7</code>	int (double)	int (double)
ceil (x)	<cmath>	Returns the smallest whole number that is not less than x: <code>ceil (56.34) = 57.0</code>	double	double
cos (x)	<cmath>	Returns the cosine of angle: x: <code>cos (0.0) = 1.0</code>	double (radians)	double
exp (x)	<cmath>	Returns e^x , where $e = 2.718$: <code>exp (1.0) = 2.71828</code>	double	double
fabs (x)	<cmath>	Returns the absolute value of its argument: <code>fabs (-5.67) = 5.67</code>	double	double

Copyright © Dr. Mohammed Hava

Electrical Engineering Department, University of Jordan

3

Function	Header File	Purpose	Parameter(s) Type	Result
floor (x)	<cmath>	Returns the largest whole number that is not greater than x: <code>floor (45.67) = 45.00</code>	double	double
islower (x)	<cctype>	Returns true if x is a lowercase letter; otherwise, it returns false ; <code>islower ('h')</code> is true	int	int
isupper (x)	<cctype>	Returns true if x is an uppercase letter; otherwise, it returns false ; <code>isupper ('K')</code> is true	int	int
pow (x, y)	<cmath>	Returns x^y ; if x is negative, y must be a whole number: <code>pow (0.16, 0.5) = 0.4</code>	double	double
sqrt (x)	<cmath>	Returns the nonnegative square root of x; x must be nonnegative: <code>sqrt (4.0) = 2.0</code>	double	double
tolower (x)	<cctype>	Returns the lowercase value of x if x is uppercase; otherwise, it returns x	int	int
toupper (x)	<cctype>	Returns the uppercase value of x if x is lowercase; otherwise, it returns x	int	int

Copyright © Dr. Mohammed Hava

Electrical Engineering Department, University of Jordan

4

C Library (more) C++ Library is as big

Function	Effect
<code>strcpy(s1, s2)</code>	Copies the string <code>s2</code> into the string variable <code>s1</code> The length of <code>s1</code> should be at least as large as <code>s2</code>
<code>strcmp(s1, s2)</code>	Returns a value < 0 if <code>s1</code> is less than <code>s2</code> Returns 0 if <code>s1</code> and <code>s2</code> are the same Returns a value > 0 if <code>s1</code> is greater than <code>s2</code>
<code>strlen(s)</code>	Returns the length of the string <code>s</code> , excluding the null character

<http://www.cplusplus.com/>

The screenshot shows the homepage of www.cplusplus.com/. It features a navigation menu on the left with links for Home, Information, Tutorial, Reference, Forum, and Headers. The main content area is divided into several sections:

- Information:** Provides general information about the C++ programming language, including non-technical documents and descriptions.
- Tutorial:** Offers a series of C++ language tutorials, including the features of the standard library, pointers, functions, and more.
- Reference:** Contains descriptions of the most important classes, functions, and objects of the Standard Library, as well as descriptive fully-functional short programs as examples.
- Forum:** A message board where members can exchange knowledge and concerns.
- Headers:** A list of C library headers, including:
 - `<assert.h>` C Diagnostic Library (header)
 - `<ctype.h>` Character handling functions (header)
 - `<errno.h>` C Errors (header)
 - `<float.h>` Characteristics of floating-point types (header)
 - `<iso646.h>` ISO 646 Alternative operator spellings (header)
 - `<limits.h>` Sizes of integral types (header)
 - `<locale.h>` C Localization library (header)
 - `<math.h>` C Numerics library (header)
 - `<setjmp.h>` Non local jumps (header)
 - `<signal.h>` C Library to handle signals (header)
 - `<stdarg.h>` Variable arguments handling (header)
 - `<stdlib.h>` C Standard definitions (header)
 - `<string.h>` C Library to perform Input/Output operations (header)
 - `<stringf.h>` C Standard General Utilities Library (header)
 - `<stringz.h>` C Strings (header)

Parameters and Return Values

```
int max(int a, int b)
{
    if(a > b)
        return a;
    else
        return b;
}
```



Copyright © Dr. Mohammed Hawa

```
int min(int a, int b)
{
    return (a < b? a : b);
}
```

Electrical Engineering Department, University of Jordan

7

```
int max(int a, int b); // prototypes
int min(int a, int b);
```

```
int main(int argc, char *argv[])
```

```
{
    int x = 17;
    int y = 5;

    int big;
    int small;

    big = max(x, y); // copy the values of x and y
    small = min(x, y); // copy the values of x and y

    printf("The maximum of %d and %d is %d\n", x, y, big);
    printf("The minimum of %d and %d is %d\n", x, y, small);

    return 0;
}
```

```
int max(int a, int b)
```

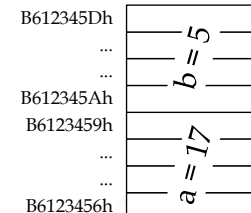
```
{
    if(a > b)
        return a;
    else
        return b;
}
```

```
int min(int a, int b)
```

```
{
    return (a < b? a : b);
}
```

Copyright © Dr. Mohammed Hawa

Pass-by-Value



Electrical Engineering Department, University of Jordan

8

Example

```
void print_hashes(int count)
{
    for (int i = 0; i < 30 - count; i++)
        std::cout << ' '; // blank

    for (int i = 0; i < count; i++)
        std::cout << ' ' << '#'; // hash

    std::cout << std::endl;
}

int main()
{
    int noOfLines = 100;

    while (noOfLines < 0 || noOfLines > 20)
    {
        std::cout << "\nEnter number of hash lines (1-20) ";
        std::cin >> noOfLines;
    }

    for (int count = 0; count < noOfLines; count++)
        print_hashes(count);

    return 0;
}
```

Enter number of hash lines (1-20) 15

```
  #
 ##
###
####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
```

Copyright © Dr. Mohammed Hawa

Electrical Engineering Department, University of Jordan 9

Pass-by-Reference: Pointers

```
void swap(double* p1, double* p2)
{
    double temp; // local variable

    temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}

int main()
{
    double x = 9.5;
    double y = 20.1;

    printf("Before: x = %f, y = %f\n", x, y);

    // swap(x, y); // wrong way (call-by-value)
    swap(&x, &y); // correct way (call-by-reference)

    printf("After: x = %f, y = %f\n", x, y);

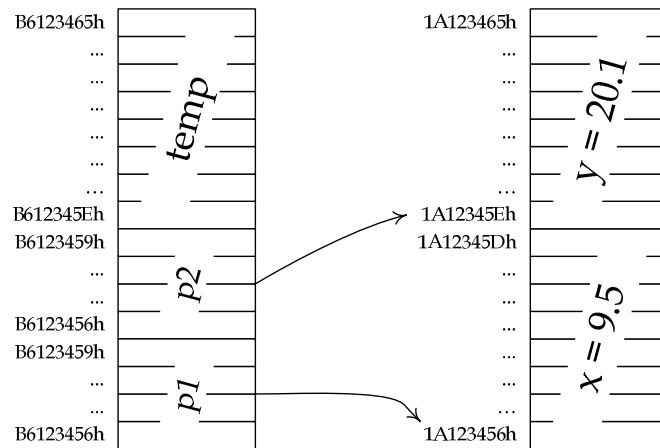
    return 0;
}
```



Copyright © Dr. Mohammed Hawa

Electrical Engineering Department, University of Jordan 10

Memory



Copyright © Dr. Mohammed Hawa

Electrical Engineering Department, University of Jordan 11

Pass-by-Reference: References

```

void swap_adv(double& r1, double& r2)
{
    double temp; // local variable

    // reference seems like a variable
    // but acts like a pointer
    temp = r1;
    r1 = r2;
    r2 = temp;
}

int main()
{
    double x = 9.5;
    double y = 20.1;

    printf("Before: x = %f, y = %f\n", x, y);

    swap_adv(x, y); // correct way (call-by-reference)

    printf("After: x = %f, y = %f\n", x, y);

    return 0;
}

```

Copyright © Dr. Mohammed Hawa

Electrical Engineering Department, University of Jordan 12

Example: Linear Search

- Linear search is also known as sequential search
- The most obvious algorithm for search
- Start at the beginning and walk to the end, testing for a match at each item
- Notice the passing of the array base address (instead of a copy of the array)
- *Homework:* Read about binary search and bubble sort in your textbook.



Copyright © Dr. Mohammed Hawa

Electrical Engineering Department, University of Jordan 13

Linear Search

```

// return the index of the element in array
// or -1 if the element does not exist
int linear_search(int array[], int size, int element)
{
    for(int i = 0; i < size; i++)
    {
        if(array[i] == element)
            return i; // index
    }

    return -1; // did not find the element
}

int main4()
{
    int my_values[10] = {11, 44, 32, 51, 7, -5, 4, 99, 100, -77};
    int index_of_32 = linear_search(my_values, 10, 32);
    int index_of_55 = linear_search(my_values, 10, 55);

    printf("The index of 32 is %d\n", index_of_32);
    printf("The index of 55 is %d\n", index_of_55);

    return 0;
}

```



Copyright © Dr. Mohammed Hawa

Electrical Engineering Department, University of Jordan 14

Homework

- Get the scores of a student in all his Tawjihi courses
- Remember each course has a different maximum (Arabic 70, Math 100)
- Calculate the student average
- Use a function called:
`normalize (score, max)`
- There is no need to handout this homework.

