LEHIGH UNIVERSITY

C Programming II

Alexander B. Pacheco
LTS Research Computing
June 3, 2015

Outline

Functions

Functions

- A function is a group of statements that together perform a task.
- Every C program has at least one function, which is main()
- Functions receive either a fixed or variable amount of arguments.
- Functions can only return one value, or return no value (void).
- In C, arguments are **passed by value** to functions
- How to return value? - **Pointers**
- Functions are defined using the following syntax:

```
return_type function_name( parameter list )
{
  body of the function
}
```

- A function **declaration** tells the compiler about a function's name, return type, and parameters.
- A function **definition** provides the actual body of the function.

Function Definition

- **Return Type:** Function's return type is the data type of the value the function returns. When there is no return value, return void.
- **Function Name:** This is the actual name of the function.
- **Parameter:** The parameter list refers to the type, order, and number of the parameters of a function. A function may contain no parameters.
- **Function Body:** The function body contains a collection of statements that define the function behavior.

```
/* function returning the max between two numbers */
int max(int i, int j)
{
  /* local variable declaration */
  int result;

  if (i > j)
    result = i;
  else
    result = j;

  return result;
}
```

Example of using a Function

```
#include <stdio.h>

/* function declaration */
int max(int i, int j);

int main() {

  /* local variable definition */
  int i = 100, j = 200, maxval;

  /* calling a function to get max value */
  maxval = max(a, b);

  printf( "Max value is : %d\n", maxval );
  return 0;

}


/* function returning the max between two numbers */
int max(int i, int j)
{
  /* local variable declaration */
  int result;

  if (i > j)
    result = i;
  else
    result = j;

  return result;
}
```

Scope Rules: Local & Global Variables I

- A scope is a region of the program where a defined variable can have its existence and beyond that variable can not be accessed.
- **Local Variables:** declared inside a function or block.

  can be used only by statements that are inside that function or block of code.

  Local variables are not known to functions outside their own.
- **Global Variables:** defined outside of a function, usually on top of the program.

  will hold their value throughout the lifetime of your program and,

  they can be accessed inside any of the functions defined for the program.
- A program can have same name for local and global variables but value of local variable inside a function will take preference.

Scope Rules: Local & Global Variables II

```c
#include <stdio.h>

/* global variable declaration */
int a = 20;

int main ()
{
  /* local variable declaration in main function */
  int a = 10;
  int b = 20;
  int c = 0;

  printf ("value of a in main() = %d\n",  a);
  c = sum( a, b);
  printf ("value of c in main() = %d\n",  c);

  return 0;
}

/* function to add two integers */
int sum(int a, int b)
{
  printf ("value of a in sum() = %d\n",  a);
  printf ("value of b in sum() = %d\n",  b);

  return a + b;
}
```

```
    value of a in main() = 10
    value of a in sum() = 10
    value of b in sum() = 20
    value of c in main() = 30
```

Initializing Local & Global Variables

- Local Variables are not initialized by the system, the programmer must initialize it.
- Global variables are automatically initialized by the system depending on the data type

Data Type	Initial Default Value
int	0
char	'\0'
float	0
double	0
pointer	NULL

- *It is a good programming practice to initialize variables properly otherwise, your program may produce unexpected results because uninitialized variables will take some garbage value already available at its memory location.*

Arrays

Pointers

Input/Output