

Functions, Part 2 of 3

Topics

- Functions That Return a Value
- Parameter Passing
- Local Variables
- Header Files

Reading

- Sections 5.1 - 5.7

Functions Can Return Values

```
/*.....  
** averageTwo - calculates and returns the average of two numbers  
** Inputs: num1 - an integer value  
**          num2 - an integer value  
** Outputs: the floating point average of num1 and num2  
**.....*/  
float averageTwo (int num1, int num2)  
{  
    float average ; /* average of the two numbers */  
  
    average = (num1 + num2) / 2.0 ;  
    return average ;  
}
```

Using averageTwo

```
#include <stdio.h>  
float averageTwo (int num1, int num2) ;  
int main ()  
{  
    float ave ;  
    int val1 = 5, val2 = 8 ;  
    ave = averageTwo (val1, val2) ;  
    printf ("The average of %d and %d is %f\n", val1, val2, ave) ;  
    return 0 ;  
}  
  
float averageTwo (int num1, int num2)  
{  
    float average ;  
  
    average = (num1 + num2) / 2.0 ;  
    return average ;  
}
```

Parameter Passing

- **Actual parameters** are the parameters that appear in the function call.

```
average = averageTwo (value1, value2) ;
```

- **Formal parameters** are the parameters that appear in the function header.

```
float averageTwo (int num1, int num2)
```

- Actual and formal parameters are matched by position. Each formal parameter receives the value of its corresponding actual parameter.

Parameter Passing (cont.)

- Corresponding actual and formal parameters do not have to have the same name, but they may.
- Corresponding actual and formal parameters must be of the same data type, with some exceptions.

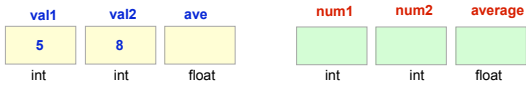
Local Variables

- Functions only “see” (have access to) their own **local variables**. This includes `main()`.
- Formal parameters are declarations of local variables. The values passed are assigned to those variables.
- Other local variables can be declared within the function body.

```
#include <stdio.h>
float averageTwo (int num1, int num2);
int main ()
{
    float ave;
    int val1 = 5, val2 = 8;
    ave = averageTwo (val1, val2);
    printf ("Ave of %d and %d is %f\n", val1, val2, ave);
    return 0;
}
```

```
float averageTwo (int num1, int num2)
{
    float average;

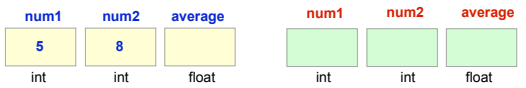
    average = (num1 + num2) / 2.0;
    return average;
}
```



```
#include <stdio.h>
float averageTwo (int num1, int num2);
int main ()
{
    float average;
    int num1 = 5, num2 = 8;
    average = averageTwo (num1, num2);
    printf ("Ave of %d and %d is %f\n", num1, num2, average);
    return 0;
}
```

```
float averageTwo (int num1, int num2)
{
    float average;

    average = (num1 + num2) / 2.0;
    return average;
}
```



Local Variables

```
#include <stdio.h>
void addOne (int number);
int main ()
{
    int num = 5;
    addOne (num);
    printf ("In main: ");
    printf ("num = %d\n", num);
    return 0;
}
void addOne (int num)
{
    num++;
    printf ("In addOne: ");
    printf ("num = %d\n", num);
}
```

Changes to Local Variables Do NOT Change Other Variables with the Same Name

```
#include <stdio.h>
void addOne (int number);
int main ()
{
    int num = 5;
    addOne (num);
    printf ("In main: ");
    printf ("num = %d\n", num);
    return 0;
}
```

num
5
int

Output:
In addOne: num = 6
In main: num = 5

```
void addOne (int num)
{
    num++;
    printf ("In addOne: ");
    printf ("num = %d\n", num);
}
```

num
6
int

Header Files

- Header files contain function prototypes for all of the functions found in the specified library.
- They also contain definitions of constants and data types used in that library.

Commonly Used Header Files

Header File	Contains Function Prototypes for:
<stdio.h>	standard input/output library functions and information used by them
<math.h>	math library functions
<stdlib.h>	conversion of numbers to text, text to numbers, memory allocation, random numbers, and other utility functions
<time.h>	manipulating the time and date
<ctype.h>	functions that test characters for certain properties and that can convert case
<string.h>	functions that manipulate character strings
others	see Chapter 5 of text

Using Header Files

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main ()
{
    float side1, side2, hypotenuse ;
    printf("Enter the lengths of the right triangle sides: ");
    scanf("%f%f", &side1, &side2);
    if ( (side1 <= 0) || (side2 <= 0) {
        exit (1);
    }
    hypotenuse = sqrt ( (side1 * side1) + (side2 * side2) );
    printf("The hypotenuse = %f\n", hypotenuse);
    return 0 ;
}
```
