

Lab 9 - CMPS 1044 Computer Science I - Functions

Lesson objectives:

- Demonstrate the use of a simple function definition and call
- Demonstrate the correct use of function prototypes
- Distinguish between use of void and int return types

Function: an independent program designed to accomplish a task (generally just ONE task)

- Calculate a value (square root, tangent, sales tax, average, maximum)
- Output (a header, an error message, a return address)

main IS a function, but today we are generally referring to programs OTHER than main.

Any function can access/call any other function (if programmed appropriately). The most common use is **main** calling other functions. That will be our emphasis in this lab.

Steps for developing and using functions:

1. Determine functions needed
2. Include prototype for each function [after #include statements but before `int main()`]
3. Include full function definition **after** main

Let us write a function called FindMax that will accept as parameters 3 integers, N1, N2, N3, then return the largest (Big) of the 3 integers. Since we will return an answer that is an integer, the type of the function must be `int`.

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

// Function Prototype; Function name is FindMax;
// Function type is int; Function parameters are 3 integers
int FindMax(int, int, int); // will determine maximum of 3 integers
// can list actual parameters as (int N1, int N2, int N3)

int main()
{
    return 0;
}

int FindMax(int N1, int N2, int N3)
{
    int Big; // not the most efficient method for finding max
    if (N1 >= N2 && N1 >= N3)
```

```

        Big = N1;
    else if (N2 >= N1 && N2 >= N3)
        Big = N2;
    else
        Big = N3;
    return Big;
}

```

To **CALL** a function, the command is placed within another function, in this case, `main`. Since `FindMax` returns one integer answer, the call must be used in such a way that **the answer is saved for future use**. For the following call, `Biggest` will be assigned the value 13.

```
Biggest = FindMax(10, 13, 7);
```

Note you can use an integer expression as an argument. If `x`, `y` & `z` are integers the following are legal.

```

Biggest = FindMax(x, y, z);
Biggest = FindMax(10, x + 7, z * 7);
Biggest = FindMax(10 - x, abs(z), x + y * (x + z));
//abs is absolute value

```

Class Exercise:

Part 1: In C++ type in the code given above into a new project. Complete `main` so that it will call `FindMax` using the command given then print `Biggest`. Correct all errors until it correctly prints 13.

Part 2: Modify `main` declaring 3 integers, `A`, `B`, and `C`. Ask the user to input 3 integers for `A`, `B`, and `C`. Then call `FindMax` to determine and print the biggest of the 3 integers.

```
Biggest = FindMax(A, B, C);
```

Part 3: Develop another function called `PrintVal` that accepts as a parameter one integer `X`. It then prints the following statement and the value of `X`. (E.G. if `X = 100`)

```
The answer is 100
```

Include the prototype for `PrintVal` immediately below the prototype for `FindMax`.

Include the complete function definition for `PrintVal` immediately below the definition of `FindMax`.

Modify `main` so that it calls `PrintVal` to print the largest value **AFTER** it receive the answer from `FindMax`.

Execute and correct until the Project runs correctly.

DOCUMENTATION of Functions:

Every function must have a set of comments describing the function as follows:

```
// Input: state function parameters and what they represent
```

```
// Processing: state task that function will accomplish
```

```
// Output: return value returned to calling function
```

For function `PrintVal`, the document should look similar to the following:

```

// Function: PrintVal
// Input:  accepts one integer parameter
// Process: print input parameter to the screen
// Output: none
// Additional comments as necessary
void PrintVal (int X)

```

The documentation can be placed:

- Above the function prototype
- Above the function definition **** Dr. Halverson's students use this location**

Part 4: Modify your code so that output is printed to a file. Since PrintVal also needs to use the output file, modify your code so that your output variable is passed as a parameter to PrintVal.

Remember that `ofstream` is a data type, so we can use a variable of that type as a parameter. Change the parameter list for PrintVal so that it includes an `ofstream` variable in both the prototype AND the definition. Since we want to make changes to the variable in the function rather than make a copy, we must pass it **by** reference. This means we must include the ampersand symbol (&) after the data type (`ofstream &` in this case). You must also modify the function call in main to pass in the name of the variable you used for your output file (e.g. `outfile`).

```

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

int FindMax(int, int, int);
void PrintVal(int, ofstream &);

int main()
{
    ofstream outfile;
    outfile.open("output.txt");
    int A, B, C, Biggest;
    cout << "Please enter 3 integers\n";
    cin >> A >> B >> C;
    Biggest = FindMax(A, B, C);
    PrintVal(Biggest, outfile);
    outfile.close();
    return 0;
}

int FindMax(int N1, int N2, int N3)
{
    int Big;

```

```
    if (N1 >= N2 && N1 >= N3)
        Big = N1;
    else if (N2 >= N1 && N2 >= N3)
        Big = N2;
    else
        Big = N3;
    return Big;
}

void PrintVal(int X, ofstream &out)
{
    out << "The answer is " << X << '\n';
}
```

Part 5: Add COMPLETE function documentation to your project then print off to turn in to your lecture instructor. Remember that you need to change the documentation for PrintVal because the parameters (inputs) have been changed and the process has been changed to print to a file.