Shyan Yen
Due April 3, 2019

PLP 5&6 - Functions, Parameters, and Scope

As always, write out the answers to these questions as though you were writing a guide for a new learner. Make your answers as clear as possible, and provide code that shows how you tested the questions and the results that you get. Post your answers and code on GitHub and email me the link.

1. What is the syntax for declaring a function in your language?

Functions must be declared above the main program or else the compiler will state that the functions "are not declared in this scope," thus the functions must be declared before using them. They are declared by its return type, function name, parameters, and ending with a semicolon. Functions are defined by stating its return type, function name, parameter list, an open bracket, function body, and a closing bracket.

2. Are there any rules about where the function has to be placed in your code file so that it can run (i.e., before main, after main, in the same file, in the same folder, etc)?

Function declarations are often declared in a separate file apart from the source file or outside the main program. For instance if a function is located in a separate .cpp file, then a header must be used. In a header, the function that you would like to use is declared and must be included in both the main file and .cpp file. The header is a separate file and is included using the #include compiler directive.

3. Does your language support recursive functions? If so, write one.

C++ does support recursive functions.

4. Can functions in your language accept multiple parameters? Can they be of different data types?

Functions in C++ accept multiple parameters. Declarations for functions have to have the same number and type of parameter as well as return type. However it is possible to return different data types by using std:: library and tuples (only works for ISO C++ 2011 standard). Following code would be above main program:

```
std::tuple<int, bool, float> multiVar()
{
   return{12, false, 1.2};
}
```

Code inside main:

```
{
    std::tuple<int, bool, float> result = multiVar();
    int value = std::get<0>(result);
    cout << "Returning multiple values of different types (int, bool, float): " << endl;
    auto [value1, value2, value3] = multiVar();
}
```

5. Can functions in your language return multiple values at the same time?

C++ does not allow for multiple values to be returned at the same time with functions. However, this can be done indirectly using references and using class and objects.

6. Declare a variable (say, x) in the main body of your program. Then declare a variable of the same name inside of a loop. Is there a conflict? Is the old variable overwritten or do you now have two variables of the same name?

All variables have scope and a storage class. The scope of the area in the program where the variable is valid. A global variable is valid from the point it is declared to the end of the program. A local variable's scope is limited to the block where it is declared and can't be accessed outside that block. The x variable gets overwritten in the loop when the variable x is stated outside the loop then inside the loop. When the same variable name is used inside of a function, there is no conflict and the x value stated inside the function is returned as that value.

7. What if the other x is inside a function?

If the other x is inside a function, the program will run and print the variable value assigned in that function.

8. Can you have variables that are globally accessible? What are the rules for creating them?

You are able to have variables that are globally accessible in C++. The global variables are declared outside of main and other functions and are accessible from all functions including the main. To access the global variable in a local scope you must use :: with the variable name. The print statement will look like cout << ::variableName << endl;

9. Are variables passed by value or by reference? (Hint: write a function that alters its input, but doesn't return it. Pass it a variable, and see if the alteration is visible in main after you call the function)

Variables in C++ are passed by reference it can also be passed by value.

10. If you run this code (or the equivalent) in your language, what is the output? What does that tell you about how the language handles assignments? (Answer in code)

```
char [] a = {'c','a','t'}
char [] b = {'d','o','g'}
a=b
b[1] = 'u'
print a
print b
```