

## CS 121 SI – Week 2 Worksheet – Includes, Namespacing, and Functions

1. What is a preprocessor directive? Give an example of a preprocessor directive and what your example does.
2. Assume you work for a company as a software migration expert and your project leader wishes to use the following program in the company's software suite:

```
#include <iostream>
#include <string>
#include <loginsys> // made-up library used for calling validUser in this exercise

using namespace std;

int main()
{
    string user_acct = "";
    int pin = -1;

    cout << "Please enter in your account user name: ";

    getline(cin, user_acct);

    cout << "Please enter in your account pin: ";

    cin >> pin;

    if ( validUser(user_acct, pin) )    // assume loginsys has this function defined
    {
        cout << "Thank you for logging into the system." << endl;

        //... other code for stuff after logging in...
    }
    else
    {
        cout << "ERROR: Invalid user data submitted. Ending program..." << endl;
        return -1;    // arbitrary return code for this program (bad login info)
    }

    return 0;    // returning 0 (historically) means no errors in the code
}
```

Unfortunately for you, the library *loginsys* reserves the keywords *cout* and *cin* for its own special purposes. To keep your job, **tweak the code so your company may seamlessly use it.**

3. Write a program that does the following:

- Takes three integer side lengths (a,b,c) from the user.
- Determine the type of triangle the user supplied (right, equilateral, or neither)
- Output the type of triangle the user supplied.
- Request again if they would like to check another.

An example output would look something similar to the following:

...

Please enter three side lengths (separated by spaces): 3 4 5

The side lengths provided form a right triangle.

Another triangle (y/n)? n

...

To save space on this paper, the program does not need to do input validation (e.g. user entered char instead of int for a side length).

4. In computer science, a function is known as a task or action. They sometimes take an input and return an output, but do not always have to do so. The format for declaring a function:

```
return_type function_identifier ( parameter_list_separated_by_commas )
{
    // statements in functions
    return return_value;
}
```

An example program:

```
#include <iostream>
```

```
double square ( double x ) // behaves identically to the math function  $f(x) = x^2$ 
{
    double result = x*x;
    return result;
}
```

```
void printHello() // a function that returns nothing (void) and no inputs (empty param list)
{
    std::cout << "Hello!\n";
}
```

```
int main() // this is also a function
{
    std::cout << "4 squared is: " << square(4.0) << std::endl; // outputs "4 squared is 16.0"
    printHello(); // outputs "Hello!"
    return 0;
}
```

Now, **convert the following math functions into C++ function declarations**. You will need to call a function from *cmath* to (easily) do the last one. Assume all variables are double-type parameters, and that the function returns a double as well.

$$f(x) = mx + b$$

$$f(x) = (y - y_0) / (x - x_0)$$

$$f(x) = x_0 + V_0 t + \frac{1}{2} a_x t^2$$

$$f(x) = \sqrt{V_0^2 + 2 a_x (x - x_0)}$$

- Looking back at problem 3: If you did not use functions where do you think they could exist? If you did use functions, list what functions you implemented.
- Compare and contrast problems 3 & 5. Is using functions in problem 3 more convenient, less convenient, or makes no difference? Explain your thoughts.