Suffix rules in Makefile

```
%.o: %.cpp %.h
    g++ -c –o $@ $<
```

- a rule that applies to all files ending in the .o suffix. The rule says that the .o file depends upon the .cpp version of the file and the .h files.
- $@ says to put the output of the compilation in the file named on the left side of the :, the $< is the first item in the dependencies list.

```
OBJ = main.o yourClass.o
a.out: $(OBJ)
    g++ -o $@ $^
```

- the special macros $@ and $^, which are the left and right sides of the :, respectively.

Useful UNIX Commands

Command	Description
man	help menu
pico	simple text editor
gcc	compiles your source code "gnu C compiler"
a.out	executes your program
ls –al	displays a long list of files "includes hidden files i.e. dot files"
pwd	prints working directory "pathname"
cd	changes directory
mkdir	creates a directory
rmdir	removes a directory
cp file1 file2	copies contents of file1 into file2
mv file1 file2	moves a file from one place to another, or change its name
rm	removes a file
more	displays a file's contents
grep	searches for a specified pattern in a file or list of files
ps	obtains the status of the active processes in the system
kill –9 pid	terminates all processes
passwd	modify a user's password
logout	terminates your session
who	display who is on the system
finger	displays the user information
date > myfile	"output redirection" saves the output of date command in myfile
cal >> myfile	"appends" calendar to myfile
cal	display a calendar and the date
wc file1	counts the number of lines, words, and characters in file1

Homework #1 (Area of a triangle)

- The area of a triangle with sides of length *a*, *b*, and *c*, can be computed by the formula

  Area = $\sqrt{s(s-a)(s-b)(s-c)}$,

  where *s = (a + b + c) / 2*.

- Write a program which will ask for the lengths of the three sides of a triangle, and output the area of the triangle.
- The outline of the program in C++ looks like this:

  [只是建議，不見得一定要遵守]

HW #1 (2)

```
#include <iostream>
using namespace std;
int main()
{
    // declaration of variables
    // prompt for and read the lengths of the sides
    // compute the semiperimeter and area
    // display the input values and the resulting area
    return 0;
}
```

HW #1 (3)

- For example, if you entered 7, 14, and 16 as the lengths, the program should compute 48.9228 as the area. The output of the program should look similar to the following:

  The area of the triangle with sides 7, 14, and 16 is 48.9228

- Test your program with the following data:

  3, 4, 5 (answer is 6.0)

  5, 12, 13 (answer is 30.0)

  7, 14, 16 (answer above)

HW #1 (4)

- Currently, the output of the program is going to the print screen window. Now, we will have the output written to a disk file in addition to the print screen window.
- Note: This should only be done **after** your program is working correctly.

- Steps to redirect output to disk file: **[如果要用 C 的方式也可以]**

1) In the section where you have the **#include** commands (at the top of the program) add the following line:

```
#include <fstream>  // For file I/O
```

HW #1 (5)

2) At the end of the variable declaration section of **main** add this statement:

ofstream outFile; // File with the results

- At the first line of the program, after all the declarations, add:

outFile.open(“hw1Output.txt”); // Open the output file

- You may choose any name as long as it is not the name of any file currently on the disk. The “ ” marks are required around the filename. This file will be placed in the same folder as your C++ source program.

HW #1 (6)

- Copy each output statement, replacing **cout** with **outFile**. For instance,

cout << “The area of the triangle is “ << area << endl;

becomes

cout << “The area of the triangle is “ << area << endl;

outFile << “The area of the triangle is “ << area << endl;

- Note you should have two output statements – one to the screen and one to the file. Change only those output statements that are actual outputs (i.e., do **not** change your prompting messages).

HW #1 (7)

- After all output operations are done, close the file:

outFile.close();

- Save your program.
- Each time you run the program, you will erase your previous output file. So, if you run the program several times, your output file will only store the output from the last time you ran the program. To resolve this issue, modify the open statement as follows:

outFile.open(“hw1Output.txt”, ios::app);

- This will cause the output to be added to the file every time you run the program.

HW #1 (8)

- Your task is to:
  1. Implement the program using C++. **[或是像 C 的 C++]**
  2. Download Java J2SE from
     http://java.sun.com/j2se/
     and rewrite the program in Java.
  3. Download Python 3 from
     https://www.python.org/downloads/
     and rewrite the program in Python.

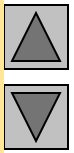
- You are required to submit a single "**makefile**" as well.

HW #1 (9)

- Learn how to set up your **debugging** environment and get experience in, for example, setting breakpoints, stepping through the execution, and evaluating a variable or an expression.

```
1  // Fig. 15.1: fig15_01.cpp
2  // Addition program
3  #include <iostream>
4
5  int main()
6  {
7     int integer1, integer2, sum;           // declaration
8
9     std::cout << "Enter first integer\n";  // prompt
10    std::cin >> integer1;                  // read an integer
11    std::cout << "Enter second integer\n"; // prompt
12    std::cin >> integer2;                  // read an integer
13    sum = integer1 + integer2;             // assignment of sum
14    std::cout << "Sum is " << sum << std::endl; // print sum
15
16    return 0;   // indicate that program ended successfully
17 }
```

```
Enter first integer
45
Enter second integer
72
Sum is 117
```



Outline

1. Load <iostream>

2. main

2.1 Initialize variables integer1, integer2, and sum

2.2 Print "Enter first integer"

2.2.1 Get input

2.3 Print "Enter second integer"

2.3.1 Get input

2.4 Add variables and put result into sum

2.5 Print "Sum is"

2.5.1 Output sum

2.6 exit (return 0)

Program Output

- The following is designed to familiarize you with the mechanics of creating, editing, compiling, and running a text-mode Java application.
- You do not have to hand it in, but you should write and run it.
- The source code in the following pages simply prompts for and accepts two numbers from the user, adds them, and displays the result.
- The file name, *Add.java* is case-sensitive and must match the class name in the program.

```
/*************************************************************
    Program to add two numbers... note that input is accepted as a
     String and then an attempt is made to convert it to a double for
     calculations. Non-numeric input is detected by the Exception
     mechanism and a default value is assigned to the value.
     *************************************************************/
import java.io.*;
import java.util.Scanner;

public class Add {
    public static void main(String args[]) {
        String amtStr;
        double num1 = 0.0, num2 = 0.0, tot = 0.0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the first number: ");
        amtStr = sc.next();
        // try to convert amt String to double for calculation
        try {  num1 = new Double(amtStr).doubleValue();  }
        catch (NumberFormatException  e) {
            System.out.println("Bad numeric input; 1st num set to 100");
            num1 = 100; }
```

```
        System.out.println("Enter the second number: ");
        amtStr = sc.next();

        try {  num2 = new Double(amtStr).doubleValue();  }
        catch (NumberFormatException  e) {
           System.out.println("Bad numeric input; 2nd num is set to 50");
           num2 = 50; }

        tot = num1 + num2;

        System.out.println("Sum is: " + tot);

      } // end main
}    // end of class Add
```

HW #1 (10)

- The **Python** program below calculates the sum of two numbers entered by the user.

```
# Store input numbers
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')

# Add two numbers
sum = float(num1) + float(num2)

# The zero argument is the name of the program file when using command line
# sum = float (sys.argv[1]) + float (sys.argv[2])

# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

HW #1 (11)

- Output:

Enter first number: 1.5

Enter second number: 6.3

The sum of 1.5 and 6.3 is 7.8

- We use the built-in function *input()* to take the input. Since, *input()* returns a string, we convert the string into number using the *float()* function. Then, the numbers are added.

The Python Standard Library

- https://docs.python.org/2/library/

- **Hint:** use the **math.sqrt** function to compute the square root. (If you are using the Python interpreter, you need to first do **import math**)