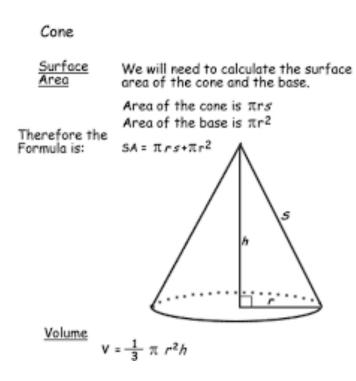
Handout-1

Recall the formulas for computing surface area and volume of a cone:



A program can have the following 3 types of errors:

- Compilation errors
- Logic errors
- Formatting errors

The program on the following page prompts a user for the height and diameter of the cone. This program has multiple errors in it.

You are to do the following:

- 1) Identify each error and specify which of the 3 types of errors it is.
- 2) Fix each error
- 3) Change the print statements so that they print only 2 decimal places, they are right aligned. Recall the maximum height and diameter are 10 meters.

```
//
//
   main.c
   APSC 160 Tutorial - Introduction + Fundamentals A
//
       Identifying errors: compilation, logic and format
//
//
    Created by Celina Berg on 2016-09-09.
//
//
    Copyright © 2016 Celina Berg. All rights reserved.
//
#include <stdio.h>
#include <math.h>
#define PI 3.14
                           /* accurate constant for PI */
int main(void) {
double w, x;
double y, z;
double r;
    r = w / 2;
printf("Please enter the diameter of your circular cone in meters: "
          " with a maximum of 10: ");
    scanf("%lf", w);
    printf("Please enter the height of your circular cone in meters: "
           " with a maximum of 10: ")
    scanf("%lf", &x);
    y = PI * r * sqrt(pow(r, 2.0) + pow(x, 2.0) + PI * pow(r, 2.0);
    z = PI * pow(r, 2.0) * x / 3;
    printf("The surface area of your circular cone is:\n %f square meters.\n" y);
    printf("The volume of your circular cone is:\n %f cubic meters.\n", z);
return 0;
```

Handout-2

Recall from in-class worksheet:

In this exercise we will investigate the use of format specifiers to control the way that output appears on the screen. When printing values of type double, format specifiers have the following form, in general:

with a + or - sign optionally appearing before the X.

Y represents a positive integer that controls the number of decimal places that are displayed. If Y is not specified, the default number of decimal places is 6. So, "%f" indicates that the number is to be printed to 6 decimal places.

X represents a positive integer that controls the width of the field in which the number is to be displayed. By "width of the field", we mean the number of consecutive characters that are used to display the output. If X is not specified, the number is displayed in a field just wide enough to print the number to the specified (or default) number of decimal places. So, "%8f" indicates that the number is to be printed in a field of width 8 characters. Note that the number is aligned on the right hand side of the field, by default. If there's a "-" sign in front of the X, the number is aligned on the left hand side of the field in which it is printed.

If there's a "+" sign in front of the X, the sign of the number is always printed. So, if the number is positive, a + sign will be printed immediately preceding the number.

Suppose the following variable declarations have been made:

```
double a = 4.5236643231; double b = -6.43864176;
```

Determine the output from each of the following printf statements. The first two are done for you – make sure you understand them before you move on.

printf statement	output											
<pre>printf("%f\n", a);</pre>	4		5	2	3	6	6	4				
<pre>printf("%f\n", b);</pre>	_	6	•	4	3	8	6	4	2			
<pre>printf("%.3f\n", a);</pre>												
printf("%.5f\n", b);												
<pre>printf("%6.2f\n", a);</pre>												
<pre>printf("%7.1f\n", b);</pre>												
printf("%-6.1f%4.1f\n", a, b);												
printf("%-7.1f%+5.1f\n", b, a);												