**Overview of *printf* statement for CIS 18B**

- Syntax:

*printf ("format string", list of variable names)*

- *printf* prints data out to screen character by character: every character in the format string gets sent out to screen one at a time

- Within the format string, there can be:

- any text character

each character is printed as shown in the format string

- escape characters

2 common ones are \t (for tab) and \n (for newline)

the escape characters are printed as they appear in the format string

- field specifications to print data stored in a variable

some common field specifications:

to print: use:

an *int*eger *%d*

a *floating point number* *%f*

a *string* *%s*

the data in the variable are printed in place of the field specification

- The list of variable names are separated by comma, if there is more than one variable

- There is a one-to-one correspondence between the field specifications ( *%s %d %f* ) in the format string

and the list of variable names, so there must be an equal number of field specifications and variable names

- Example:

If var1 is 5 and str1 is “CIS”, then: printf (“var1 has %d and str1 has %s\n”, var1, str1)

will print to screen: var1 has 5 and str1 has CIS

- When printing floating point numbers, it is useful to format the number of digits after a decimal point.

To do this, instead of using *%f*, you can use: *%.nf* for n digits after the decimal point

*printf* will do the appropriate rounding to the number of digits specified

- Example:

If var1 is 3.14159, then: printf (“var1 is %.2f\n”, var1)

will print: var1 is 3.14

and: printf (“var1 is %.4f\n”, var1)

will print: var1 is 3.1416