PROGRAMMING WITH MATLAB

WEEK 13

FILE INPUT AND OUTPUT

INPUT/OUTPUT

- Input statements read in values from the default or standard input device.
- In most systems the default input device is the keyboard.
- The input expression therefore reads the values entered by the user.
- The simplest input function in MATLAB is input.

Syntax: x = input(prompt); displays the text in prompt and waits for the user to input a value and press the Return key.

str = input(prompt,'s') returns the entered text

```
>> x = input('Please enter the first value: ')
Please enter the first value: 5
x =
     5
>> y = x^3
y =
   125
```

INPUT/OUTPUT

- Output statements display strings and the results of expressions.
- The simplest output
- function in MATLAB is disp.

Syntax: disp(x) displays the value of variable x without printing the variable name

```
>> x = [1 3 11];

>> disp(x)

    1     3    11

>> s = 'It is rainy today.';

>> disp(s)

It is rainy today.
```

INPUT/OUTPUT

- Formatted output can be printed to the screen using the fprintf function.
- The simplest output
- function in MATLAB is disp.

Syntax: fprintf(formatSpec,a1,...,an) formats data and displays the results on the screen

```
>> a1 = [1.35, 1119];
>> a2 = [7.1, 1317];
>> formatSpec = 'x is %2.2f meters and %5.3f kg\n';
>> fprintf(formatSpec, a1,a2)
x is 1.35 meters and 1119.000 kg
x is 7.10 meters and 1317.000 kg
```

INPUT/OUTPUT

- A formatting operator starts with a percent sign, %, and ends with a conversion character.

Value Type	Conversion	Explanation
Signed integer	%d or %i	Base 10
Unsigned integer	%u	Base 10
	%o	Base 8
	%x	Base 16
Floating-point number	%f	Fixed-point notation (Use a precision operator to specify the number of digits after the decimal point.)
	%e	Exponential notation
Characters or strings	%c	Single character
	%s	Character vector or string array

INPUT/OUTPUT

- There are many different file types, which use different filename extensions.
- The MATLAB has functions that can read and write data from different file types such as spreadsheets (.xls)
- Files are opened with the fopen function. By default, the fopen function opens a file for reading. For another mode, a permission string is used to specify which mode (e.g., writing or appending).

Syntax: fileID = fopen(filename,permission), opens the file with the type of access specified by permission.

filename = fopen(fileID), eturns the file name that a previous call to fopen used when it opened the file specified by fileID.

The fopen function returns -1 if it is not successful in opening the file.

INPUT/OUTPUT

Permission, file access type.

Type	Explanation
‘r’	Open file for reading (default)
‘w’	Open or create new file for writing. Discard existing contents, if any.
‘a’	Open or create new file for writing. Append data to the end of the file.
‘r+’	Open file for reading and writing.
‘w+’	Open or create new file for reading and writing. Discard existing contents, if any.
‘a+’	Open or create new file for reading and writing. Append data to the end of the file.

INPUT/OUTPUT

- Files should be closed when the program has finished reading from or writing to them.
- The function that performs this is the fclose function, which returns 0 if the file close was successful, or -1 if not

Syntax: fclose(fileID) closes an open file. fileID is an integer file identifier obtained from fopen.

fclose('all') closes all open files. Specify the literal all as a character vector or a string scalar.

status = fclose(...) returns a status of 0 when the close operation is successful. Otherwise, it returns -1.

INPUT/OUTPUT

- The MATLAB functions xlswrite and xlsread will write to and read from Microsoft Excel spreadsheet files.

Syntax: xlswrite(filename,A) writes matrix A to the first worksheet in the Microsoft Excel spreadsheet workbook filename starting at cell A1.

xlswrite(filename,A,sheet) writes to the specified worksheet.

xlswrite(filename,A,xlRange) writes to the rectangular region specified by xlRange in the first worksheet of the workbook. Use Excel range syntax, such as 'A1:C3'.

```
>> filename = ‘myData.xlsx';

>> A = [7 11 -13 17 19 0 -23];

>> xlswrite(filename,A)
```

INPUT/OUTPUT

Syntax: num = xlsread(filename) reads the first worksheet in the Microsoft Excel spreadsheet workbook named filename and returns the numeric data in a matrix.

num = xlsread(filename,sheet) reads the specified worksheet.

num = xlsread(filename,xlRange) reads from the specified range of the first worksheet in the workbook. Use Excel range syntax, such as 'A1:C3'.

INPUT/OUTPUT

values = {1, 3, 5 ; 7, 11, 13 ; ‘a’, ‘b’, ‘c’};

headers = {'First','Second','Third'};

xlswrite('myFile.xlsx',[headers; values]);

Sheet1 of myFile.xlsx contains:

First	Second	Third
1	3	5
7	11	13
a	b	c

INPUT/OUTPUT

Read numeric data from the first worksheet.

```
>> filename = 'myFile.xlsx';

A = xlsread(filename)

A =

     1     3     5

     7    11    13
```

INPUT/OUTPUT

Request the numeric data, text data, and combined data from the Excel file

```
>> [num,txt,raw] = xlsread('myFile.xlsx')

num =

     1     3     5

     7    11    13

txt =

  4×3 cell array

    'First'    'Second'    'Third'

    ''         ''          ''

    ''         ''          ''

    'a'        'b'         'c'

raw =

  4×3 cell array

    'First'    'Second'    'Third'

    [    1]    [     3]    [    5]

    [    7]    [    11]    [   13]

    'a'        'b'         'c'
```

INPUT/OUTPUT

- The save command can be used to write variables to a file, or to append variables to a mat-file. By default, the save function writes to a mat-file.

Syntax: save(filename) saves all variables from the current workspace in a MATLAB formatted binary file (MAT-file) called filename. If filename exists, save overwrites the file.

save(filename,variables) saves only the variables or fields of a structure array specified by variables.

save(filename,variables,'-append') adds new variables to an existing file. If a variable already exists in a MAT-file, then save overwrites it. The variables argument is optional.

INPUT/OUTPUT

- To variables from file into workspace:

Syntax: load(filename) loads data from filename.

load(filename,variables) loads the specified variables from the MAT-file, filename.

load(filename,'-ascii') treats filename as an ASCII file, regardless of the file extension.

load(filename,'-mat') treats filename as a MAT-file, regardless of the file extension.