(***Introduction to Mathematica: Part I.**

Mathematica is a software package that lets you do mathematics and computation,analyze data,develop algorithms,do simulation and modeling,and produce graphical displays and graphical user interfaces.This section will
introduce you to how to use Mathematica within the scope of physics laboratory experiments.In this instruction, a command in Mathematica is given near the In[...] line (you use Shift-Enter to activate the command); the number of the In[...] line is generated automatically upon each evaluation. Notes between (*...*) are just comments.

Installation of Mathematica on your own computer.
If you wish to have Mathematica software installed on your own computer,you can download it from NJIT
Information Services& Technology Department
https://ist.njit.edu/mathematica-njit*)

In[1]:=
```
(*Getting Started. To run Mathematica on a PC,double-click on the Mathematica icon.
     Help.1) if you know the exact name command,
but want to refresh which arguments it requires,use ? . E.g.,*)
?Sin
```

Out[1]=
Symbol
Sin[*z*] gives the sine of *z*.

In[2]:=
```
(* 2) if you approximately know the spelling,use ? with * for the unknown part,e.g.*)
?*ParametricPlot*
```

Out[2]=
System`
ParametricPlot ParametricPlot3D

In[3]:=
```
(*and then you click on the required command for further info*)
```

In[4]:=
```
(*3) if you have no idea-use Help button

  Note: all Mathematica commands and constants start with a Capital;
your own can start with any*)
```

In[5]:=
```
(* Ending Mathematica Session.

    Exit the notebook; save it if you need

   Inputting Commands.
To input a command and execute it,just type it  and press Shift-Enter”
in the keyboard.For example,to run simple calculation of 1+1
*)
1 + 1
```

Out[5]= 2

In[6]:= `% * 4 (*will multiply the answer by 4; % means the previous answer*)`

Out[6]= 8

(*you are now ready to use Mathematica as a simple calculator. Everything is standard but a few useful extras:

Frequent type-and space-saving commands:

1)as above, % uses the last output as input, E.g.

In[12]:=Sin[1.]

Out[12]=0.841471

In[13]:=ArcSin[%]

Out[13]=1.

2) space-can be used instead of * for multiplication:

In[1]:=2 2

Out[1]=4

Do not need space if number in front of a symbol:

In[2]:=2x

Out[2]=2 x

(useful to keep close to hand-written formulas)

but will keep a symbol unchanged if the number is after without a space

In[3]:=x2,

Out[3]=x2

so, don't forget space if you intend multiplication:

In[4]:=x 2

Out[4]=2 x

3) ; will not produce an output on the screen (but can work with it further!)
In[5]:=Factorial[1000]; (*N gives approximate numerical value*)
In[6]:=N[%]

Out[6]=$4.023872600770938\ 10^{2567}$

Brackets.
()–as in regular math
[]–argument(s) of ANY function:Sin[x], Plot[x^3, {x,-1,1}],etc. Can be empty,e.g. RandomInteger[] with output 0 or 1
{}–lists, e.g.{1,2,3} or {a,b,c}

Equal signs.
= - immediate assignment (standard)
:=- delayed. E.g.: *)

```
In[7]:= r := RandomInteger[]; Table[r, {5}]
Out[7]= {0, 0, 0, 1, 1}

In[8]:= (*Compare to *)r = RandomInteger[]; Table[r, {5}]
Out[8]= {1, 1, 1, 1, 1}

In[9]:= (*==-logical,e.g.Solve[x^2==4,x]*)

In[10]:= Solve[x^2 == 4, x]
Out[10]=
{{x → -2}, {x → 2}}

In[11]:= Sin[x]^2 + Cos[x]^2 == 1 // Simplify
Out[11]=
True

In[12]:= (*Inputting Vectors and (will not need immediately) Matrices.
A vector is an ordered list of numbers in curly brackets separated by
 comas (not spaces!). There is no difference in Mathematica between row
 vector and column vector. Often, we will label long vectors as "lists" ,
keeping the word "vectors" for physical objects in 2D and 3D. E.g. :*)

In[13]:= x = {-1, 1, 2, 4, 6}
Out[13]=
{-1, 1, 2, 4, 6}

In[14]:= (*Suppose that you want to create a vector (list) of values ranging from 1 to 10. Here’
 s how to do it without typing each number;
you can use a more universal Table or the specific Range commands*)
```

```
In[15]:= Table[i, {i, 10}]
Out[15]=
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In[16]:= Range[10](*gives the same*)
Out[16]=
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In[17]:= (*The default
increment is 1 and so is the initial value.The increment can be specified,
together with initial value, as *)

In[18]:= Table[i, {i, 1, 9, 2}]
Out[18]=
{1, 3, 5, 7, 9}

In[19]:= Range[1, 9, 2](*gives the same*)
Out[19]=
{1, 3, 5, 7, 9}

In[20]:= (*Increments can be fractional or negative. The elements of the vector X can be
extracted as X⟦1⟧,X⟦2⟧,etc. Note double squarebrackets! For example:*)

In[21]:= X = {3, 2, 4, 6, 20}
Out[21]=
{3, 2, 4, 6, 20}

In[22]:= X⟦1⟧
Out[22]=
3

In[23]:= X⟦3⟧
Out[23]=
4

In[24]:= (*A matrix is a "nested" list:*)

In[25]:= A = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}}
Out[25]=
{{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}}

In[26]:= (*to get a human view use MatrixForm:*)

In[27]:= A // MatrixForm
Out[27]//MatrixForm=
```

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

```
In[28]:= (*The math operation (+,-, *, /) also can work on matrices. and
 vectors. Make sure the dimenssions match! Some examples*)
```

```
In[29]:= A = {2, 3, 5, 10, 3}
Out[29]=
{2, 3, 5, 10, 3}

In[30]:= B = {1, 3, 4, 5, 4} * 2
Out[30]=
{2, 6, 8, 10, 8}

In[31]:= A * 3
Out[31]=
{6, 9, 15, 30, 9}

In[32]:= A + B
Out[32]=
{4, 9, 13, 20, 11}

In[33]:= A.B (*dot product*)
Out[33]=
186

In[34]:=

In[35]:= (*Some points should be made here.*)

In[36]:= A^2 // MatrixForm (*squares every element of a matrix*)
Out[36]//MatrixForm=
```

$$\begin{pmatrix} 4 \\ 9 \\ 25 \\ 100 \\ 9 \end{pmatrix}$$

```
In[37]:= (*to multiply a square matrix by itself use MatrixPower or . (dot) product *)

In[38]:= A = {{1, 2}, {2, 1}}
Out[38]=
{{1, 2}, {2, 1}}

In[39]:= A // MatrixForm
Out[39]//MatrixForm=
```

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

```
In[40]:= MatrixPower[A, 2] // MatrixForm
Out[40]//MatrixForm=
```

$$\begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix}$$

```
In[41]:= A.A // MatrixForm (*same*)
Out[41]//MatrixForm=
```

$$\begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix}$$

In[42]:=

In[43]:=
```
(*Algebraic or Symbolic Computation
For conducting symbolic computations,you do not need to declare anything,
but it can be  useful  to clear the variables that you plan to use in the
calculation. For example,
if you want to calculate (x-y)*(x-y)^2 in Mathematica,see the below:*)
```

In[44]:=
```
Clear[x, y];
```

In[45]:=
```
(x - y) * (x - y) ^2
```

Out[45]=

$(x - y)^3$

In[46]:=
```
(*To differentiate function of x^3 in
 Mathematica (D is the command for differentiation),*)
```

In[47]:=
```
D[x^3, x]
```

Out[47]=

$3 x^2$

In[48]:=
```
(*Inputting Mathematical Functions
In Mathematica you will use both built-
 in functions and functions that you create yourself.
   Built-in Functions
Mathematica has many built-in functions.These include Sqrt (for square root),
Cos (for cosine),Sin (for sine),Tan
(for tangent),Log (for natural logarithm,ln),
Exp (for exponential function),and ArcTan (for arctangent), as well
as more specialized mathematical functions. Mathematica also has several built-
 in mathematical constants,including Pi (the number π),
I (the complex number i=√-1), and Infinity.  E.g.:*)
Log[Exp[2]]
```

Out[48]=

2

In[49]:=
```
Log[10, 100]
```

Out[49]=

2

In[50]:=
```
Sin[Pi](*calculates exactly*)
```

Out[50]=

0

In[51]:=
```
(*or, approximately, but with very high accuracy *)pi = N[Pi, 25]
```

Out[51]=

3.141592653589793238462643

In[52]:=
```
Sin[pi]
```
Out[52]=
$0. \times 10^{-25}$

In[53]:=
```
(*User-Defined Functions.
You will learn two methods to define your
  own functions in Mathematica. See the below examples:*)
```
In[54]:=
```
(*The 1st way, mostly useful for simple formulas: *)
```
In[55]:=
```
Clear[f, x]; (*an immediate assignment of a function, but x, the variable,
must be free to change and take various values in the future*)
```
In[56]:=
```
f = x^2 + x
```
Out[56]=
$x + x^2$

In[57]:=
```
(*Now we can replace x by any value, using /. *)
```
In[58]:=
```
f /. x → 2
```
Out[58]=
6

In[59]:=
```
f /. x → -2
```
Out[59]=
2

In[60]:=
```
(*the function can operate on lists (vectors):*)
f /. x → {1, 3, 4}
```
Out[60]=
{2, 12, 20}

In[61]:=
```
(*The 2nd way is using delayed assignement; you do not have to Clear *)
```
In[62]:=
```
g[x_] := x^2 + x
  (*the underscore _ indicates a slot in which you can put anything instead of x *)
```
In[63]:=
```
g[2]
```
Out[63]=
6

In[64]:=
```
g[Sin[x]]
```
Out[64]=
$\text{Sin}[x] + \text{Sin}[x]^2$

In[65]:=
```
g[z]
```
Out[65]=
$z + z^2$

In[66]:=
```
g /@ {1, 2, 3} (* this is how the function g is applied to
   a list (vector). The /@ applies g to every element of the list;
  note: no square brackets at this stage*)
```
Out[66]=
```
{2, 6, 12}
```
In[67]:=
```
(*For the function with multi-variables,
there is nothing new in Mathematica, just write:*)
```
In[68]:=
```
Clear[f, x, y];
f = x^2 +
   y^2 (* for the 1st way of defining a function and then replace x an y using /.  :*)
```
Out[68]=

$x^2 + y^2$

In[69]:=
```
f /. {x → 1, y → 2}
```
Out[69]=
```
5
```
In[70]:=
```
g[x_, y_] := x^2 + y^2
```
In[71]:=
```
g[1, 2]
```
Out[71]=
```
5
```
In[72]:=
```
(*Solving Equations*)
```
In[73]:=
```
Clear[x]; Solve[x^2 - 3 * x + 1 == 0, x] (*Note the double == *)
```
Out[73]=

$\left\{\left\{x \to \frac{1}{2}\left(3 - \sqrt{5}\right)\right\}, \left\{x \to \frac{1}{2}\left(3 + \sqrt{5}\right)\right\}\right\}$

In[74]:=
```
N[%, 7] (*gives numeriical approximation of the exact result to 7 digits of accuracy*)
```
Out[74]=
```
{{x → 0.3819660}, {x → 2.618034}}
```
In[75]:=
```
(*or, if I am happy with default build-in accuracy, just use //N after  the result*)
```
In[76]:=
```
Solve[x^2 - 3 * x + 1 == 0, x] // N
```
Out[76]=
```
{{x → 0.381966}, {x → 2.61803}}
```
In[77]:=
```
(*The command Solve can solve higher-degree polynomial equations,
    as well as many other types of equations.It can
also solve equations involving more than one variable.You should
specify  which variable(s) to solve for by giving a list of those. For example,
    try to type Solve(2*x-log(y)=1,y] to
solve 2x-log y=1 for y in terms of x.*)
```

In[78]:=
```
Solve[2 * x - Log[y] == 1, y]
```

Out[78]=

$\{\{y \to e^{-1+2x} \text{ if } -\pi < 2\,\text{Im}[x] \le \pi\}\}$

In[79]:=
```
FullSimplify[%, Assumptions → x > 0](*we need to tell Mathematica that x is real*)
```

Out[79]=

$\{\{y \to e^{-1+2x}\}\}$

In[80]:=
```
(*You can specify more than one equation as well by giving a
 list of equations followed by the list of unknowns.For example:*)
```

In[81]:=
```
Solve[{x^2 - y == 2, y - 2 * x == 5}, {x, y}]
```

Out[81]=

$\{\{x \to 1 - 2\sqrt{2}, y \to 7 - 4\sqrt{2}\}, \{x \to 1 + 2\sqrt{2}, y \to 7 + 4\sqrt{2}\}\}$

In[82]:=
```
% // N
```

Out[82]=

{{x → -1.82843, y → 1.34315}, {x → 3.82843, y → 12.6569}}

In[83]:=
```
(*for example, the second solution can be extracted using [[2]], as for any list*)%[[2]]
```

Out[83]=

{x → 3.82843, y → 12.6569}

In[84]:=
```
(*Performing Calculus
Differentiation
The command to differentiate symbolic expressions is D, i.e. df/dx is D{f,x]. The
 syntax for second derivatives is D[f,{x,2}],and for nth derivatives, D[f,{x,n}].*)
```

In[85]:=
```
D[x^3, x]
```

Out[85]=

$3x^2$

In[86]:=
```
(*or*)f = x^3; D[f, x]
```

Out[86]=

$3x^2$

In[87]:=
```
D[f, {x, 2}]
```

Out[87]=

$6x$

In[88]:=
```
(*The command D can also compute partial derivatives of a functions
 with several variables,as in D[x*y,y],and to do multiple partials with
 respect to mixed variables you can use D repeatedly:*)D[D[x * y / z, x], y]
```

Out[88]=

$\frac{1}{z}$

In[89]:=
```
(*Integration
Mathematics can compute definite and
 indefinite integrals.Here is an indefinite integral:*)
```

In[90]:=
```
Integrate[x^2, x]
```

Out[90]=

$$\frac{x^3}{3}$$

In[91]:=
```
(*and here is a definite integral:*)
Integrate[x^2, {x, 0, 1}]
```

Out[91]=

$$\frac{1}{3}$$

In[92]:=
```
(*Mathematica can also do multiple integrals.The
 following command computes the double integral:*)
```

In[93]:=
```
Integrate[x^2 + y^2, {x, 0, 1}, {y, 0, 1}]
```

Out[93]=

$$\frac{2}{3}$$

In[94]:=
```
(*Data Analysis*)
```

In[95]:=
```
(*We have a data of (1,3,5,7,1,6,4,12).Let
 's try to sort the data and then find maximum,minimum,
  and median value among the data,followed by mean value and standard deviation.First,
  you have to assign variable to the data:*)
x = {1, 3, 5, 7, 1, 6, 4, 12};
```

In[96]:=
```
Sort[x]
```

Out[96]=

{1, 1, 3, 4, 5, 6, 7, 12}

In[97]:=
```
Max[x]
```

Out[97]=

12

In[98]:=
```
Min[x]
```

Out[98]=

1

In[99]:=
```
Median[x]
```

Out[99]=

$$\frac{9}{2}$$

In[100]:=

```
Mean[x]
```

Out[100]=

$$\frac{39}{8}$$

In[101]:=

```
% // N
```

Out[101]=

4.875

In[102]:=

```
StandardDeviation[x]
```

Out[102]=

$$\frac{\sqrt{\frac{727}{14}}}{2}$$

In[103]:=

```
% // N
```

Out[103]=

3.60307

In[104]:=

```
Length[x]
```

Out[104]=

8

In[105]:=

```
Plus @@ x (*sum of all elements*)
```

Out[105]=

39

In[106]:=

```
Mean[x] Length[x](*should be same*)
```

Out[106]=

39