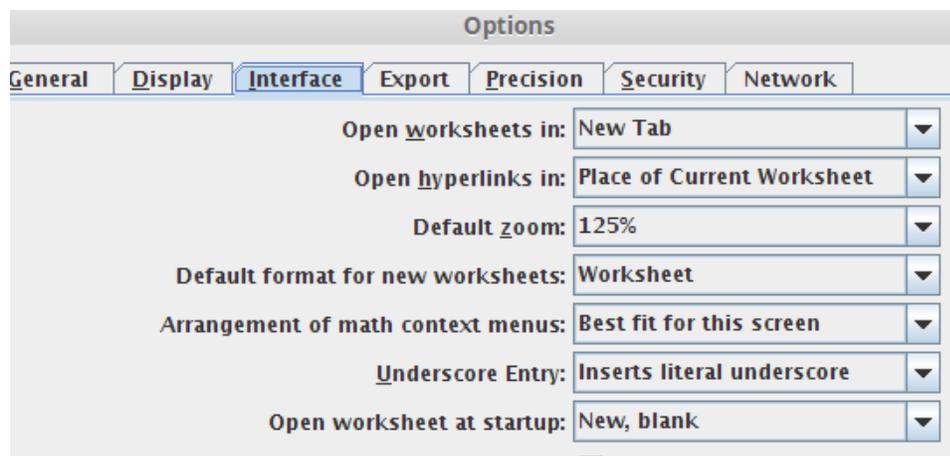
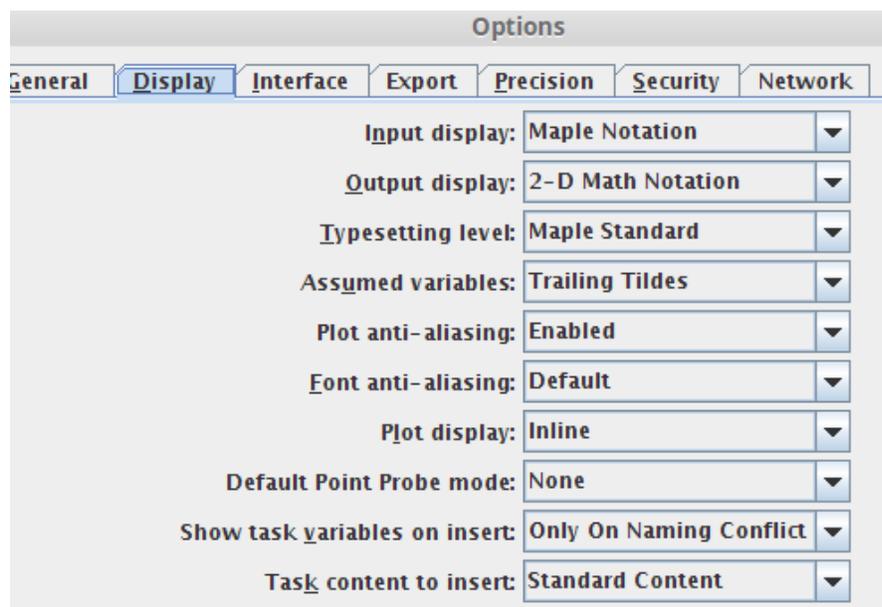


Intro Maple Lab 2280 Spring 2018

Your **Maple** typing lesson is to be done in the computer lab classroom LCB 115. If you missed it, then do it on your own in the Math Center computer lab, lowest floor between buildings JWB and LCB. Report by email how it worked out.

Maple requires some setup to be useful as a programming tool. Expected is RED print for keystrokes and BLUE print for Maple's formatted output. The recommendations:

Open the Tools menu and select the **Options** item. A popup window will appear. Change the Maple option tabs **Display** and **Interface** to match the following screen shots:



To make these settings your defaults for all future use of Maple on this computer, click the **Apply Globally** button at the bottom of the **Options** tab.

restart:

```
# intro Computer Lab
```

```
# Spring 2018
```

```
# Use the pound sign "#" to make a comment. What follows # on that line is a comment.
```

```
# Use comments to put your name and the name of the assignment at the top of the worksheet.
```

```
# Use comments to introduce each problem. or major step.
```

```
# End each non-comment line with a semicolon (;) or a colon (:)
```

```
# A semicolon causes BLUE echo and a colon causes no echo.
```

```

;
2+2;
3*5;
3^6*5/2;
7*(9+11);
# Use the assignment operator "==" (colon equals) to assign a value to a symbol.
x:=2;
x+3;
# Define a column vector with angle brackets.
v:=<1,2,3>;
w:=<1,0,0>;
#Use a period "." for dot products.
v.w;
v.v;
# Define a matrix using column vectors separated by a bar "|".
M:=<v|w|<1,1,0>>;
# Matrix multiplication and Matrix times vector also use a
period.
M.M;
M.v;
N:=<<1,2,3,4>|<5,6,7,8>|<9,10,11,12>>;
N.M;
M.N; # This should not work, the matrices have incompatible shapes.
# Matrices can also be entered in rows using square brackets with the "Matrix" function.
P:=Matrix([[1,2,3],[4,5,6],[7,8,9]]);
# Alternate entry by columns, then transpose the matrix.
P1:=<1,2,3|4,5,6|7,8,9>^+;
# Linear combinations of matrices can be computed using expected math-style operations.
2*P;
P-M;
5*P-3*M;
# The percent sign "%" recalls the result of the previous computation.
#This is useful for multi-step computations like Gaussian Elimination.
# Use Gaussian Elimination to reduce the following matrix to upper triangular form.
Q:=<<1,2,1>|<2,3,4>|<1,1,1>>;
E1:=<<1,-2,-1>|<0,1,0>|<0,0,1>>;
Q1:=%.Q; # Left multiple by Elimination matrix E1
E2:=<<1,0,0>|<0,1,2>|<0,0,1>>;
Q2:=%.Q1; # Left multiple by Elimination matrix E2
# "%" recalls the result of the last computation.
# "%%" recalls the result of the second to last computation.
# "%%" recalls the result of the third to last computation.
# You can use up to 3 percent signs at a time.
%,%,%%%;
# Using % is not recommended for rookies. Use instead LABELS, like (10).
#
# The matrix Q has three non-zero pivots, so it is invertible.
# You can ask for the inverse using two different notations.
# An answer check is inverse(Q) times Q = identity matrix.
Q^(-1); 1/Q; %.Q;

```

```

# We can also do symbolic computations.
# undefine symbols before starting ..
a:='a':b:='b':c:='c':
A:=Matrix([[a[1,1],a[1,2]],[a[2,1],a[2,2]]]);
B:=Matrix([[b[1,1],b[1,2]],[b[2,1],b[2,2]]]);
C:=Matrix([[c[1,1],c[1,2]],[c[2,1],c[2,2]]]);
# Verify associativity of matrix multiplication.
(A.B).C-A.(B.C);
# We expect to get the zero matrix.
# To encourage the maple engine to simplify algebraic expressions, use:
simplify(%);
# It's good practice to do elimination step by step, but of
# course we would like Maple to do it for us.
# Let's load the maple library for linear algebra, as follows.
# Only do this once per session. The colon is used remove BLUE printout.
with(LinearAlgebra):
# Perform Elimination, showing only the answer, no steps.
# We choose the system Qx=b, where b:=<1,2,3>:
b:=<1,2,3>:
Q;
Aug:=<Q|b>;
GaussianElimination(Aug);
ReducedRowEchelonForm(Aug);
#
# Elimination steps with LinearAlgebra functions. Definitions:
combo:=(a,s,t,c)->LinearAlgebra[RowOperation](a,[t,s],c);
swap:=(a,s,t)->LinearAlgebra[RowOperation](a,[t,s]);
mult:=(a,t,c)->LinearAlgebra[RowOperation](a,t,c);
A1:=<Q|b>;          # Do 9-10 steps with combo, swap, mult.
A2:=combo(A1,1,2,-2); # Invent the other steps.
# This is a good way to do homework problems. Answer check:
ReducedRowEchelonForm(A1);
#
# DIFFERENTIAL EQUATIONS
#
# Maple solves differential equations using dsolve:

de := diff( y(x), x ) = y(x) - sin(x);
sol:= dsolve( de, y(x));

# Initial conditions give a unique solution that can be graphed.

de := diff( y(x), x ) = y(x) - sin(x);
# Also: f:=(x,y)->y-sin(x); de:=D(y)(x) = f(x,y(x));
# But: diff(y,x)=y-sin(x) is a SYNTAX ERROR
ic := y(0)=1;
sol:= dsolve( [de,ic], y(x));
# Both syntax [de,ic] and syntax {de,ic} will work.

# To graph the solution, extract the expression after the equal sign

```

```

# of the answer with Maple function rhs().

u:=rhs(sol);
plot(u,x=0..10);

# Try ?dsolve for more information and examples.

# DIRECTION FIELDS and PHASE PLOTS
# The DEplot command from the DEtools package can plot
# a direction field and threaded curves.
# EXAMPLE. Field only for  $y'=y-\sin(x)$ 

with(DEtools):
de := diff(y(x),x) =y(x) -sin(x);
DEplot(de,y(x),x=-3..3,y=-3..3);

#EXAMPLE. Direction field and threaded curves for  $y'=y-\sin(x)$ 

ics:=[ [y(0)=0], [y(0)=1], [y(0)=3] ];
# Set of initial conditions. A double list.
# Identical Result: ics:=[[0,0],[0,1],[0,3]]:
opts:=color='black',linecolor='red',arrows='line',dirgrid=[30,30]:
DEplot(de,y(x),x=-3..3,y=-3..3,opts,ics);

# SECOND ORDER ORDINARY SCALAR EQUATIONS

# Solve  $y'' + y = 0$  with initial conditions

de1 := diff( y(x),x,x) + y(x) = 0;
ic := y(0)=1,D(y)(0)=0;
dsolve( [de1,ic], y(x) );

# Solve  $y'' + y = \sin(x)$  with initial conditions

de2 := diff( y(x),x,x) + y(x) = sin(x);
ic := y(0)=1,D(y)(0)=0;
dsolve( [de2,ic], y(x) );

# End of lab0-intro

```