Table of Contents

..... 1
_____ 2
INITIALIZATION 2
_____ 2
CALCULATIONS 2
_____ 2
COMMAND WINDOW OUTPUTS 2
_____ 4
ACADEMIC INTEGRITY STATEMENT 4


```
function [numTerms, cosVal, absDiff] =
 PS08_taylor_cos_koike_zhou823(xVal, tol)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% This program is designed to calculate the approximate value of the
% cosine of a given number of summing an unknown number of terms in
% the series.
%
% Function Call
% [numTerms, cosVal, absDiff] = PS08_taylor_cos_koike_zhou823(xVal,
 tol);
%
% Input Arguments
% 1. x_value: an arbitrary value of x
% 2. tol: the tolerance of the while loop
%
% Output Arguments
% 1. numTerms: the number of terms in the Taylor series
% 2. cosVal: the calculated value of cosine using Taylor series
% 3. absDiff: the absolute difference between the Taylor series
 computed
% value of cosx and the value returned by MATLAB's built-in cosine
% function.
%
% Assignment Information
%   Assignment:   PS 08, Problem 1
%   Team ID:   002-08
%   Paired Partner:  Tomoki Koike, koike@purdue.edu
%   Paired Partner:  Yi Zhou, zhou823@purdue.edu
%   Contributor:  Name, login@purdue [repeat for each]
%   Our contributor(s) helped us:
%     [ ] understand the assignment expectations without
%         telling us how they will approach it.
%     [ ] understand different ways to think about a solution
%         without helping us plan our solution.
%     [ ] think through the meaning of a specific error or
%         bug present in our code without looking at our code.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIALIZATION

```
% the array for academic integrity statement
nameArray = ["Tomoki Koike" "Yi Zhou"];

% the valid conditions for the
% x_value
xInvalid = isstring(xVal) == 1 | ischar(xVal) == 1 | isscalar(xVal) ==
 0;
tolInvalid = ~(0<=tol & tol<=1);
```

CALCULATIONS

```
%initialize outputs
numTerms = -99;
cosVal = -99;
absDiff = -99;

% the main sturctural operation

    if xInvalid
        fprintf("\nError! Invalid x value\n");
    elseif tolInvalid
        fprintf("\nError! Invalid tolerance\n");
    else
        cosVal = 1;
        cosNth = 1;
        n=1;
        while abs(cosNth)>tol
            cosNth = (-1)^n * xVal^(2*n) / factorial(2*n);
            cosVal = cosVal + cosNth;
            n = n + 1;
        end
    absDiff = abs(cos(xVal)-cosVal);
    numTerms = n;
    end
```

COMMAND WINDOW OUTPUTS

```
%xVal=2;
%tol=0.01;
```

```
%[numTerms, cosVal, absDiff] = PS08_taylor_cos_koike_zhou823(xVal,
 tol)

%numTerms =

%     5


%cosVal =

%   -0.4159


%absDiff =

%   2.7382e-04
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%xVal="a";
%tol=0.1;
%[numTerms, cosVal, absDiff] = PS08_taylor_cos_koike_zhou823(xVal,
 tol)

%Error! Invalid x value

%numTerms =

%   -99


%cosVal =

%   -99


%absDiff =

%   -99
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%xVal=12;
%tol=3;
%[numTerms, cosVal, absDiff] = PS08_taylor_cos_koike_zhou823(xVal,
 tol)

%Error! Invalid tolerance

%numTerms =

%   -99


%cosVal =
```

```
%  -99


%absDiff =

%   -99
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%xVal=20;
%tol=0.0005;
%[numTerms, cosVal, absDiff] = PS08_taylor_cos_koike_zhou823(xVal,
 tol)

%numTerms =

%    31


%cosVal =

%    0.4081


%absDiff =

%   1.3327e-05
```

ACADEMIC INTEGRITY STATEMENT

```
% Call your academic integrity statement here
PS07_academic_integrity_koike(nameArray)
```

We are submitting code that is our own original work. We have not used source code, either modified or unmodified, obtained from any unauthorized source. Neither have we provided access to our code to any peer or unauthorized source. Signed,
<Tomoki Koike>
<Yi Zhou>

Published with MATLAB® R2018a