Table of Contents

```
function [counter, Taylor_cos, difference] =
 PS08_taylor_jchapla_rbehl(x,tolerance)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% This function finds the value of the Taylor Series of cosine for a
 given
% value and below a certain tolerance. It also calculates the regular
% cosine of that same number then finds the difference between the
 two.
%
% Function Call
% function [num_term, Taylor_cos, difference] =
 PS08_taylor_jchapla_rbehl(x,tolerance)
%
% Input Arguments
% 1. x //user input value for x
% 2. tolerance //user input tolerance
%
% Output Arguments
% 1. counter //the number of terms that the loop goes through
% 2. Taylor_cos //approximated value for cos using the taylor series
% 3. difference //difference between approximated value and actual
 value of
% cos
%
% Assignment Information
%   Assignment:   PS 08, Problem 1
%   Team ID:   008-14
%   Paired Partner:  John Chapla, jchapla@purdue.edu
%   Paired Partner:  Ranjan Behl, rbehl@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.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Initializes variables that will be used for calculations

INITIALIZATION

initial value of the series

```
k = 0;
% initializes counter
counter = 0;
% initializes the absolute value of N
absN = 0;
% initializes the
Taylor_cos = 0;
```

If/else structure and while loop for executing taylor series estimation

CALCULATIONS

if/else structure that tests if the inputs are invalid and also carries out the calculations if they are valid

```
if (isscalar(x) == 0) %invalid x
    counter = -99;
    Taylor_cos = -99;
    difference = -99;
    fprintf("x value is invalid, enter a scalar\n");
elseif (tolerance <= 0 || tolerance >= 1) %invalid tolerance
    counter = -99;
    Taylor_cos = -99;
    difference = -99;
    fprintf("tolerance is invalid, must be between 0 and 1\n");
else %valid x and tolerance
    termVal = ((-1).^k * x.^(2 * k)) / factorial(2 * k);
    Taylor_cos = Taylor_cos + termVal;
    absN = abs(termVal);
        while (absN > tolerance) %while loop for taylor series
            k = k + 1;
            termVal = ((-1).^k * x.^(2 * k)) / factorial(2 * k);
            Taylor_cos = Taylor_cos + termVal;
            absN = abs(termVal);
            counter = counter + 1;
        end
    difference = abs(Taylor_cos - cos(x));
end
```

```matlab
% Prints the results from the if/else and while loop
fprintf(" Number of Terms = %d\n Taylor Value of cos(x) =
 %0.10f\n Difference between cos(x) and Taylor approximation =
 %0.10f",counter,Taylor_cos,difference);
```

```
 Number of Terms = 6
 Taylor Value of cos(x) = -0.6507594463
 Difference between cos(x) and Taylor approximation = 0.0028841745
```

Shows command window outputs

COMMAND WINDOW OUTPUTS

Valid test case PS08_taylor_jchapla_rbehl(0.5,0.05) Number of Terms = 2 Taylor Value of cos(x) = 0.8776041667 Difference between cos(x) and Taylor approximation = 0.0000216048

```matlab
% Invalid x test case
% PS08_taylor_jchapla_rbehl([1.2,0.01])
% x value is invalid, enter a scalar
%  Number of Terms = -99
%  Taylor Value of cos(x) = -99.0000000000
%  Difference between cos(x) and Taylor approximation = -99.0000000000

% Invalid tolerance test case
% PS08_taylor_jchapla_rbehl(0.5,3)
% tolerance is invalid, must be between 0 and 1
%  Number of Terms = -99
%  Taylor Value of cos(x) = -99.0000000000
%  Difference between cos(x) and Taylor approximation = -99.0000000000
```

Calls academic integrity statement

ACADEMIC INTEGRITY STATEMENT

```matlab
PS07_integrity_rbehl(["John Chapla","Ranjan Behl"]);
```

```
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,
 <John Chapla>
 <Ranjan Behl>
```

Published with MATLAB® R2018b