
Table of Contents

.....	1
.....	2
INITIALIZATION	2
.....	2
CALCULATIONS	2
.....	2
COMMAND WINDOW OUTPUTS	2
.....	3
ACADEMIC INTEGRITY STATEMENT	3

```
function [termNum,tayCos, difCos] =
    PS08_taylor_cos_wpopovic_ehotson(x,tol)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% This function calculates the cosine of an imputed value to an
% imputed
% tolerance using a taylor series. It outputs to how many terms the
% taylor
% series was iterated, the taylor approximation, and the absolute
% value of
% the difference between the approximation and the actual value.
%
% Function Call
% [termNum,tayCos, difCos] = PS08_taylor_cos_wpopovic_ehotson(x,tol)
%
% Input Arguments
% 1. x %value we are finding the cos of
% 2. tol % tolerance for taylor series
%
% Output Arguments
% 1. termNum %the number of terms used in the taylor series
% 2. tayCos % the cos(x) as approximated by the taylor series
% 3. difCos % absolute value of the difference between the taylor
% series
% approximation and the cos found by the matlab function
%
% Assignment Information
% Assignment: PS 08, Problem 1
% Team ID: 009-01
% Paired Partner: William Popovich, wpopovic@purdue.edu
% Paired Partner: Ethan Hotson, ehotson@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

```
%preset outputs to invalid results
termNum = -99;
tayCos = -99;
difCos = -99;
```

CALCULATIONS

```
if isscalar(x) == 0 %is x invlaid
    fprintf('Error invalid x. Must be a scalar.\n')
elseif tol <= 0 || tol >= 1 %is tolerance invalid
    fprintf('Error invalid tolerance. Must be between 0 and 1, not
    inclusive.\n')
else
    tayCos = 1; %initial value of taylor approximation
    absTerm = 1; %the absolute value of the current term of the taylor
    series
    termNum = 1; %term counter
    while absTerm > tol
        termN = (((-1) ^ termNum) * (x ^ (2 * termNum))) / factorial(2
        * termNum); %nth term of series
        tayCos = tayCos + termN; %add new term to approximation
        absTerm = abs(termN);
        termNum = termNum + 1; %increment counter
    end
    difCos = abs(cos(x) - tayCos); %calculate absolute value of
    difference between taylor cosine approximation and cos function
end

Not enough input arguments.

Error in PS08_taylor_cos_wpopovic_ehotson (line 50)
if isscalar(x) == 0 %is x invlaid
```

COMMAND WINDOW OUTPUTS

```
% [termNum,tayCos, difCos] = PS08_taylor_cos_wpopovic_ehotson(.5,.05)
%
% termNum =
%
```

```

%      3
%
%
% tayCos =
%
%      0.8776
%
%
% difCos =
%
%      2.1605e-05
%
% [termNum,tayCos, difCos] =
%   PS08_taylor_cos_wpopovic_ehotson([0,2,5],.05)
% Error invalid x. Must be a scalar.
%
% termNum =
%
%      -99
%
%
% tayCos =
%
%      -99
%
%
% difCos =
%
%      -99
%
% [termNum,tayCos, difCos] = PS08_taylor_cos_wpopovic_ehotson(.5,2)
% Error invalid tolerance. Must be between 0 and 1, not inclusive.
%
% termNum =
%
%      -99
%
%
% tayCos =
%
%      -99
%
%
% difCos =
%
%      -99

```

ACADEMIC INTEGRITY STATEMENT

% Call your academic integrity statement here

Published with MATLAB® R2018b