

```

1 %% Creates Object with static methods to define functions in a class
2 classdef set_1
3     methods(Static)
4
5         %% taylor function fpr hyperbolic cosine
6         % @param take the input value and desired error
7         % @return the vomputed value of x
8         function[n] = taylor_cosh(x, err)
9             res = 0;
10            diff = 1;
11            n = 0;
12            expected = cosh(x);
13            while (diff > err)
14
15                res = res + ((x)^(2*n) ) / (factorial(2*n));
16
17                n = n + 1;
18
19                diff = abs(expected - res);
20
21            end
22        end
23        %% Fib RecurrSION Definition
24        function[N] = fib_rec(N)
25            if (N < 3)
26                N = 1;
27            else
28
29                N = set_1.fib_rec(N-1) + set_1.fib_rec(N-2);
30            end
31        end
32
33        function[j, fiblis] = fib_list(N)
34            fiblis(1) = 1;
35            fiblis(2) = 1;
36            for n=1: (N - 2)
37                fiblis(n + 2) = fiblis(n + 1) + fiblis(n);
38
39            end
40
41            j = fiblis(N);
42        end
43        %% Calculates the ratio
44        % returns 1, unless it is requested with two or more terms
45        % to me the function is not zero, but one given the definition
46        % provided
47        function[sig] = ratio(N)
48            [~, seq] = (set_1.fib_list(N));
49            sig = 1;
50            for n=1:(N - 1)
51                sig = sig + ( (-1)^(n+1) ) / (seq(n) * seq(n+1));
52            end
53        end
54    end
55 end

```

