MATH 4610 w/ Joe Koebbe
Fall 2021

By Spencer Wheeler A02242077

1. I think I will choice to write code for the class in python, to respond to "it's only a bunny" I wrote the line, print("He's not just a bunny, he smells fear, and will rip your throat out!!"). This is all you need to to have this print "He's not just a bunny, he smells fear, and will rip your throat out!! to the console.

2. I created a README.md file on GitHub, as I didn't select that option when I created the repository. So I created it and then add the description and links that was required from me.

3. $f'(a) = \frac{f(a+h)-f(a-h)}{2h} \approx \frac{1}{2h}(f(a)+f'(a)h+\frac{1}{2}f''(a)h^2+\frac{1}{6}f'''(\xi_1)h^3 - f(a)+f'(a)h-\frac{1}{2}f''(a)h^2+\frac{1}{6}f'''(\xi_2)h^3$
   $= \frac{1}{2h}(2f'(a)h + \frac{1}{6}f'''(\xi_3)h^3) = f'(a) + \frac{1}{12}f'''(\xi_3)h^2$
   error $= |f'(a) - (f'(a) + \frac{1}{12}f'''(\xi_3)h^2)| = \frac{1}{12}f'''(\xi_3)h^2$
   error $\leqslant ch^2$

4. $f''(a) = \frac{f(x+h)-2f(x)+f(x-h)}{h^2}$
   $\approx 1/h^2(f(a) + f'(a)h + 1/2f''(a)h^2 + 1/6f'''(\xi_1)h^3 + f(a) + f'(a)(-h) + 1/2f''(a)h^2 - 1/6f'''(\xi_2) - 2f(a))$
   $= f''(a) + \frac{1}{6}f'''(\xi_3)h$
   error $\leqslant ch$

   I wrote code to approximate the second derivative of Cosine, that can be found here,

   `https://github.com/spencerwheeler2077/math4610/blob/main/SecDerivative.py`

   When run it will produce this produce this table in the console.

h-value	error of approximate second derivative.
1e-0	0.03354335418516324
1e-1	0.00034667345475336564
1e-2	3.467876013296678e-06
1e-3	3.4640095514237856e-08
1e-4	1.9541062379335727e-08
1e-5	2.802191542139454e-07
1e-6	0.00014626922009641774
1e-7	0.022391258179794205
1e-8	0.6940761880780137
1e-9	55.09500439471066
1e-10	5550.698976289233
1e-11	555111.0961657414
1e-12	0.4161468365471424
1e-13	5551115122.709634
1e-14	1665334536938.1506
1e-15	277555756156288.72
1e-16	0.4161468365471424

5. In my search for more information about finite difference approximations I found this one resource that showed that we can expand the Taylor series out further, we can create higher order approximations. I would give some of the examples this pdf gives, but I don't quite understand what it is saying and it would be very long to type in. It seems like there are many different ways to get even more accurate approximations.

   `https://www.dam.brown.edu/people/alcyew/handouts/numdiff.pdf` This is the website that I visited.