I, __________________________, pledge that this exam is completely my own work, and that I did not take, borrow or steal any portions from any other person. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of the San Diego State University Policies.

________________________________________
Signature

**Be sure to show all your work or include a copy of your programs.**

1. Consider the function given by

$$f(x) = 2e^{-2x} - 2\cos(2x) + 4\,\sin(x).$$

a. Write a Taylor polynomial about $x_0 = 0$ of degree 3, $P_3(x)$, and include the remainder term $R_3(x)$ for $x \in [0, 1]$.

b. Use the remainder term to get an upper bound on the error in the approximation, $P_3(x)$, for $x \in [0, 1]$. Find the absolute and relative error between $f(1)$ and $P_3(1)$.

c. One of the roots of this function is $x = 0$. Write Newton's method for finding this root. Show your iterations to a tolerance of $10^{-5}$, starting with $x_0 = 0.3$. What is the rate of convergence for this iteration procedure? Explain.

d. Write down a scheme that converges more rapidly than your Newton's method. Show your iterations for this scheme to a tolerance of $10^{-8}$, starting with $x_0 = 0.3$. What is its rate of convergence for this iteration procedure? If applicable, explain why the last iterate does not illustrate this same rate of convergence.

e. Create a graph of $f(x)$ with points clearly showing the location for each of your roots. Use the **secant method** and **Newton's method** to find the other roots of $f(x) = 0$ for $x \in (0, 8]$. For both methods use $x_0 = 2.5$ and $x_0 = 7.0$, then for the secant method use $x_1 = 2.4$ and $x_1 = 6.9$, respectively. Show your iterations to a tolerance of $10^{-9}$. The Cauchy convergence condition from class,

$$\lim_{n\to\infty} \frac{|x_{n+1} - x_n|}{|x_n - x_{n-1}|^{\alpha}} = L, \tag{1}$$

becomes a line after taking logarithms:

$$\ln\left(|x_{n+1} - x_n|\right) = \ln(L) + \alpha \ln\left(|x_n - x_{n-1}|\right).$$

With your iteration data find the best linear fit in the formula above and determine the best slope, $\alpha$, and the value of $L$. Provide a well-labeled graph showing the data points and this best fitting line. With this graphical support explain the rate of convergence for each fixed point with each method. Would your rate of convergence change based on your initial guess? Explain.

f. **Basins of Attraction:** We noted in class that it is hard to determine how far away from a root an initial estimate $x_0$ can be in order to converge to the root $x_*$. Take our interval $I = [0, 8]$ and select initial points at every 0.1, $x_0 = 0, 0.1, 0.2, ..., 7.9, 8$. Create a program to see where the Newton iterates for each of these $x_0$ values converge in less than `Nmax = 50` iterates with `tol = 1e-6`. Your solution should have a table of the $x_0$ values and the Root value that was found. (Make your table with 6 or 8 columns to not be too long.) Determine the intervals where the $x_0$ values converge to the closest $x_*$. Note when the convergence is to another root and not the closest $x_*$. Write a paragraph about what you observe from the results of your program and give some explanations for your observations. (Hint: The graph of the original function and the geometric interpretation of Newton's method may help explain your observations.)

2. a. Write a program that generates the iterative sequence

$$y_n = a\, y_{n-1} + b\, y_{n-2}, \quad n \geq 2, \quad a, b \in \mathbb{R}. \tag{2}$$

Your program must take $a$, $b$, $y_0$, $y_1$, and the maximum value of $N$ as input, and it must produce a plot of the sequence for $0 \leq n \leq N$ with axis labeled semi-log plots, (MatLab `semilogy`). For $N = 20$ and the parameter values:

1. $a = 1$, $b = 1.2$, $y_0 = 2$, $y_1 = 1$,
2. $a = 0.3$, $b = 0.6$, $y_0 = 2$, $y_1 = 6$,

use your program to generate two plots. Explain the behavior you see in the graphs. In particular, find the slope of the lines in your plots in the large $N$ limit both computationally and analytically. (Hint: In order to get the analytic result, you need to use a guess for the solution of the form

$$y_n = \lambda^n.$$

Then use this $y_n$ in Eqn. (2), and solve for $\lambda$.)

b. Take your program in Part a with $a = 1$, $b = 1.2$, $y_0 = 2$, $y_1 = 1$. Modify the code to add every third term (starting with $y_0$), $(y_0 + y_3 + y_6 + y_9 + ...)$ as long as the sum is less than $M$ (an input to your program). The program should output the values of $3n$, $y_{3n}$, and the sum $y_3 + y_6 + y_9 + ... + y_{3n} < M$. Run this program for $M = 1,000,000$. Create a table that has columns with $n = 0, 3, 6, 9, ...$, $y_0, y_3, y_6, y_9, ...$, and $y_0, y_0 + y_3, y_0 + y_3 + y_6, y_0 + y_3 + y_6 + y_9, ...$ showing the output of your program.

3. The Greek mathematician Archimedes estimated the number $\pi$ by approximating the circumference of a circle of diameter 1 by the perimeter of both inscribed and circumscribed polygons. The perimeter, $t_n$, of the circumscribed regular polygon with $2^n$ sides can be given by the recursive formula ($t_n > \pi$):

$$t_{n+1} = \frac{2^{n+1}\left(\sqrt{1 + \left(\frac{t_n}{2^n}\right)^2} - 1\right)}{\left(\frac{t_n}{2^n}\right)}, \qquad t_2 = 4.$$

a. Write a MatLab program to calculate $t_3$ to $t_{30}$. Describe what goes wrong with your calculation and why this occurred.

b. Use some algebra to correct the problem and recompute $t_3$ to $t_{30}$.

4. To solve the quadratic equation

$$ax^2 + bx + c = 0,$$

we usually use the "classic" formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{3}$$

Explain when this formula suffers from catastrophic cancellation errors on a finite-precision computer.

b. Assume that you have a 4-digit computer (rounding last digit), and the quadratic equation has the parameters $a = 1.000$, $b = -71.82$, and $c = 0.2741$. Find the two real roots for this quadratic equation using your 4-digit computer. (For partial credit you must show intermediate steps.) Find the actual roots and determine the percent error between the actual roots and the roots found with 4-digit rounding.

c. Derive and explain why and when it is better to use the expression

$$x = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}. \tag{4}$$

Write a program that computes the roots of the quadratic equation by switching between the "classic" equation (3) and the new formula (4). Clearly explain how you choose the criteria for switching and why your approach is superior to strictly using the classic equation.

d. Repeat Part b with your 4-digit computer using the new formula (4) for the roots of the quadratic equation. (For partial credit you must show intermediate steps.) Use the actual value for these roots and determine the percent error between the actual roots and the roots found with 4-digit rounding and formula (4).

5. a. Consider the functions

$$f(x) = 1.8\, e^{0.7x} \qquad \text{and} \qquad g(x) = 5.3\, x^4.$$

These curves intersect when $f(x) = g(x)$ or $F(x) = f(x) - g(x) = 0$. Create programs for the bisection, secant, and Newton's methods to find all roots of $F(x) = 0$ to an accuracy of $10^{-5}$ for bisection and $10^{-8}$ for secant and Newton's methods. Describe how you found reasonable initial points for determining your roots. Give your starting values and give the number of iterations until you converge to a root. What criterion do you use for convergence? Briefly discuss the efficiency and stability of each of your programs. Note which program would be your preferred method. Produce a graph or graphs to clearly illustrate all points of intersection for $f(x)$ and $g(x)$, and give both the $x$ and $y$-values for the points of intersection.

b. Compare the rate of convergence, $\alpha$, of the three methods from Eqn. (1) using log / log-plots. Your solution must include the graphs showing convergence and give the values of $\alpha$ and $L$. Explain all details clearly, using the iterations from the negative root found in Part a.