C200 Programming Assignment № 2

Functions & Learning About Homework
Fall 2025

Dr. M.M. Dalkilic
Computer Science
School of Informatics, Computing, and Engineering

Indiana University, Bloomington, IN, USA

September 11, 2025

The purpose of this homework is two-fold:

(1) Understand the eco-system

1. We push the homework to your GitHub
2. You pull to your laptop and work on it continually pushing it back
3. We pull it from GitHub at the due date

(2) Get some experience with Python sessions and simple coding. The end of the document has helpful key shortcuts.

(3) You'll work on a single file `hw2.py`

☐ All modules that are required will be imported at the beginning of the document; you do not need any others
☐ The homework takes only a few hours–we give you a week to make the process easier; so, there is more than ample time to complete it; postponing until a few hours until it's due is the least effective approach to learn to program;
☐ It's best to uncomment only one function as you work, then commenting it back and moving to the next one.
☐ You must complete this before **Friday, September 19, 2025 11:00PM EST**. You will submit your work by pushing your code to your GitHub repository and by uploading your completed python file to the Autograder on Gradescope, only the **first 10 attempts will be considered and the highest will be used to grade.** Please remember that:

- You <u>never</u> turn anything in on canvas.
- You <u>never</u> manually upload files to your repository using GitHub's *Upload files* tool.
- Your code must neither raise an exception nor be the product of anyone or anything else
- You <u>never</u> change any function's name or signature

- If you have a partner, you both must put comments at the top of your respective homework name, email. You both always submit individually
- Questions about the models are given as thinking exercises–nothing to actually turn-in

Important

Each function has an empty body with `pass`. The first problem will be done in lab as an example. Some of these problems were taken or inspired by the excellent introductory texts *Applied Calculus* by Tan, 2005, *Thinking Mathematically* 5^{th} *ed.* by Blitzer, 2011.

Problem 1: Volume of a Cone

The volume of a cone with radius r and height h is:

$$c(r,h) = \frac{1}{3}\pi r^2 h \tag{1}$$

For example, 2 cm radius and 5 cm height,

$$c(2,5) \approx 20.94\ cm^3 \tag{2}$$

For 3 cm radius and 7 cm height

$$c(3,7) \approx 65.97\ cm^3 \tag{3}$$

Deliverables Problem 1

- Complete the cone volume function in the file hw2.py.
- You must use math.pi from the math module.
- Round the answer to 2 decimal places. Use round(x,y) to round x to y decimal places. For example,

```
>>> round(1.252,2)  #rounds to two places
1.25
>>> round(1.252,1)  #rounds to 1 place
1.3
>>> round(1.252,0)  #rounds to 0 place
1.0
```

As an example, here's code with rounding two 1 place in the function:

```
def foo(x,y,z):
    ans_ = round((x + y + z)/3,1)
    return ans_

if __name__ == "__main__":

    """
    If you want to do some of your own testing in this file,
    please put any print statements you want to try in
    this if statement.

    You **do not** have to put anything here  """

    # my test
    print(foo(200,300,500))
```

has output:

```
333.3
```

Problem 2: Oxygen Content of a Pond

The oxygen content t days after organic waste has been dumped into a pond is given by:

$$f(t) = 100\frac{t^2 + 10t + 100}{t^2 + 20t + 100} \tag{4}$$

percent of its normal level. For example,

$$f(0) = 100 \tag{5}$$

$$f(10) = 75 \tag{6}$$

Deliverables Problem 2

- Complete the oxygen content function in the file hw2.py.
- Round the value you function returns to 2 decimal places.

Problem 3: TV Viewing Patterns

According to A.C. Nielsen Co., the percent of U.S. households waching television during the weekdays (about a decade ago) starting at 4:00PM for eight hours is modeled as $P(t)$:

$$P(t) \quad = \quad 0.01354t^4 - 0.49375t^3 + 2.58333t^2 + 3.8t + 31.60704 \tag{7}$$

if $0 \le t \le 8$ where $t = 0$ corresponds to 4:00P. For example,

$$P(0) \quad = \quad 31.61\% \tag{8}$$

$$P(3) \quad = \quad 54.02\% \tag{9}$$

$$P(8) \quad = \quad 30.0\% \tag{10}$$

Deliverables Problem 3

- Complete tv percent function in the file hw2.py.
- Round the answer to 2 decimal places.

Problem 4: Toxic Waste

A city's main well was recently found to be contaminated with trichloroethylene, a cancer-causing chemical, as a result of an abandoned chemical dump leaching chemicals into the water. A proposal submitted to city council members indicates that the cost, measured in millions of dollars to remove x% of the toxic pollutant is given by:

$$cost(x) = \frac{0.5x}{100 - x} \tag{11}$$

for $0 < x < 100$. For eample, 50%, 70%, and 90% cost

$$cost(50) = \$.5 \text{ million} \tag{12}$$
$$cost(70) \approx \$1.17 \text{ million} \tag{13}$$
$$cost(90) = \$4.5 \text{ million} \tag{14}$$

Deliverables Problem 4

- Complete the percent cost function in the file hw2.py.
- Round the answer to 2 decimal places.

Problem 5: Cowling's Rule

Cowling's rule is a method for calculating pediatric drug dosages. If a denotes the adult dosage(in milligrams) and t is the age of the child (in years), then the child's dosage is given by:

$$D(t,a) \quad = \quad \frac{t+1}{24}a \tag{15}$$

For example, if $a = 500 \text{ mg}$ and $t = 4 \text{ years}$, then

$$D(t,a) \quad = \quad 104.17 \text{ mg} \tag{16}$$

$$\tag{17}$$

Deliverables Problem 5

- Complete Cowling's rule function in the file hw2.py.
- Round the answer to 2 decimal places.

Problem 6: Flu Outbreak

During a flu outbreak in a school of 1000 children, the number of infected children, I, was expressed in terms of the number of susceptible (but still healthy) children, S, by:

$$I(S) \quad = \quad \lceil 192 \log_2(\frac{S}{762}) - S + 763 \rceil \tag{18}$$

For example,

$$I(100) \quad = \quad 101 \text{ students} \tag{19}$$

$$I(300) \quad = \quad 205 \text{ students} \tag{20}$$

Deliverables Problem 6

- The math module is imported for this problem.
- Complete the I function in the file hw2.py.
- For this problem, you'll need to make your own input/outputs.
- You may get negative values for some inputs values that you provide to the function, and that's fine.
- To remind you, $\lceil x \rceil = y$ is the smallest integer y such that $x \leq y$.
- Use **math.ceil()** from the math module discussed in lecture.

Problem 7: Average Cost

Let $C(q)$ be the cost of producing q items. A company's cost function is (in \$) given by

$$C(q) = 0.01q^2 - 0.6q + 1000 \tag{21}$$

The **average cost** $A(q)$ is given by

$$A(q) = \frac{C(q)}{q} \tag{22}$$

For example,

$$A(1) = \$1012.41 \tag{23}$$
$$A(10) = \$108.0 \tag{24}$$
$$A(316) = \$825.12 \tag{25}$$

Deliverables Problem 7

- Complete both functions C, A in the file hw2.py
- For those interested, more is given at the end about how to use average cost and marginal cost for business decisions. None of the extra information is required–just encouraged for those who are interested.

Problem 8: Sales Model

A company contracted by you has built a sales model that returns the number of plastic ducks sold ($\times 10^3$)

$$hh(t) = \left\lfloor \frac{532}{1 + 869e^{-1.33t}} \right\rfloor \tag{26}$$

For $0 \leq t \leq 11$ where t is months. The floor function $\lfloor\ \rfloor$ is in the math module named math.floor() For example,

$$hh(0) = 0 \text{ ducks} \tag{27}$$
$$hh(5) = 250 \text{ ducks} \tag{28}$$
$$hh(10) = 531 \text{ ducks} \tag{29}$$
$$hh(11) = 531 \text{ ducks} \tag{30}$$

Deliverables Problem 8

- The math module is imported for this problem.
- Complete the hh function in the file hw2.py.
- Round your answer to the nearest integer using math.floor()

Problem 9: Throwing a Stone

A stone is thrown straight up from the roof of an 80 ft building with an initial velocity of $64\,\text{ft/sec}$. The height (in feet) of the stone at any time t seconds is given by:

$$height(t) = -16t^2 + 64t + 80 \tag{31}$$

The rock will hit the ground after 5 seconds:

$$height(5) = 0 \text{ feet} \tag{32}$$

Deliverables Problem 9

- Complete the $height$ function in the file hw2.py
- Round the answer to 2 decimal places.

Problem 10: Treating Heart Attacks

According to the American Heart Association, the treatment benefit for heart attacks depends on the time (hours) until treatment and is described by:

$$B(t) \quad = \quad \frac{0.44t^4 + 700}{0.1t^4 + 7}, \text{ for } (0 \leq t \leq 24) \tag{33}$$

Deliverables Problem 10

- Complete the B function in the file hw2.py.
- For this problem, you'll need to make your own input/outputs.
- Round the answer to 2 decimal places.

Problem 11: Average Rate of Change

Given a function $f(x)$ and two points, a, b where $a < b$, the average rate of change is defined as:

$$arc(f, a, b) \quad = \frac{f(b)-f(a)}{b-a} \tag{34}$$

A model of per capita cigarette consumption beginning in 1980 is:

$$cig(t) \quad = \quad 3870(0.970)^t \tag{35}$$

where t is the number of years since 1980. Then, the average rate of change in per capita cigarette consumption from 1985 to 2005 is

$$arc(cig, 1985-1980, 2005-1980) \quad = \quad \frac{cig(25)-cig(5)}{25-5} \approx -76 \tag{36}$$

A model of kilograms of methamphetamine seized along the border of the U.S. in year y is:

$$mk(y) \quad = \quad 151.500000y^2 - 606377.300265y + 606755991.065880 \tag{37}$$

The average rate of change of seizures from 2000 to 2004 is:

$$arc(mk, 2000, 2004) \quad = \quad \frac{mk(2004)-mk(2000)}{2004-2001} \approx 228 \tag{38}$$

Deliverables Problem 11

- Complete `arc(f,a,b)`,`cig(t)`, `mk(y)`
- For `arc`, `mk` use the floor function
- For `cig` use the ceiling function

Problem 12: Sinking Fund

This model describes a “sinking fund." Money is periodically set aside until a date is reached. One important kind is retirement–money that will be available when you stop working. You must know the payment amount P, the number of times a year you make the payment n, the number of years you make the payment t, and the interest rate r. For example,

- If you pay $22,000 once a year for seven years that has 6% compounded annually, you'll have, at the end $184,664.43.
- If you make $500 monthly that has 4% (about the current best rate) compounded monthly for 20 years, you'll have $183,387.31
- If $1,200 is deposited quarterly with 8% compounded quarterly for 10 years, you'll have $72,482.38

The sinking fund is:

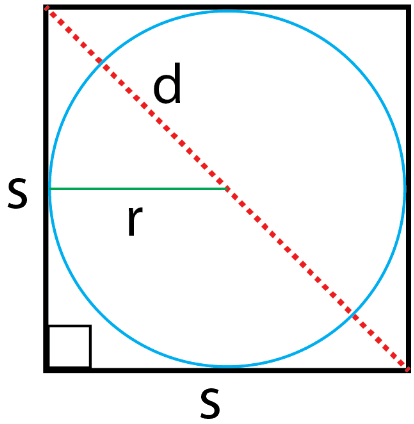
$$R(P, r, n, t) \quad = \quad P\left[\frac{(1+\frac{r}{n})^{nt}-1}{\frac{r}{n}}\right] \tag{39}$$

Deliverables Problem 12

- Complete the R function in the file hw2.py
- Round to two decimal places.

Problem 13: Geometry

You are given two functions to implement. The first takes the length of a side of a square and returns the diagonal (no units). The second takes a diagonal of a square and returns the area of the largest circle that can be inscribed in the square. A figure is shown below.



The side is s, the diagonal (red dotted line) is d, the radius (green) is r, and the circle blue. The easiest solution involves using the Pythagorean Theorem. Here is a run:

```
s = 10
a  = circle_area(square_diagonal(s))
print(a)
```

has output:

```
78.54
```

where we have rounded to two decimal places. As you can see, there is function composition: the output of the first function is the input to the second.

Deliverables Problem 13

- Complete the two functions in the file hw2.py
- You can create some more solutions to verify your implelmentation is correct.
- Round to two decimal places.

Problem 14: Distance of Kicked Football

A kicked ball will travel some distance as a function of the angle $\theta\,\mathrm{deg}$ (degrees) of impact and velocity $v\,\frac{\mathrm{m}}{\mathrm{s}}$ (meters per second) as

$$dk(v,\theta) \quad = \quad \frac{v^2}{g}\sin(2\theta) \tag{40}$$

where g is acceleration due to gravity $\approx 9.8\,\frac{\mathrm{m}}{\mathrm{s}^2}$. Python's math module includes trigonometric functions, but these use radians, a dimensionless unit describing angular measurement that provide many computational and mathematical benefits over degrees. The simple relationship to remember is:

$$2\pi\,\text{radian} \quad = \quad 360\,\text{degree} \tag{41}$$

The math module includes a function to convert between the two and we show them here.

```
>>> import math
>>> d = 90 #degrees
>>> d * ((2*math.pi)/(360)), math.radians(d)
(1.5707963267948966, 1.5707963267948966)
>>> d = 40 #degrees
>>> d * ((2*math.pi)/(360)), math.radians(d)
(0.6981317007977318, 0.6981317007977318)
```

Since we're using meters, we want to convert our final answer to feet. We know:

$$1\,\text{m} \quad = \quad 3.2808399\,\text{ft} \tag{42}$$

For example, if the ball is kicked at $20\,\frac{\mathrm{m}}{\mathrm{s}}$ at an angle 40°, then the distance traveled (in feet) is

$$dk(20,40) \quad = \quad 131.88 \tag{43}$$

when rounding to two decimal places.

Deliverables Problem 14

- Complete the `dk(v,theta)` function
- Round to two decimal places.
- You must use acceleration due to gravity and conversion from meters to feet. The best approach would be to declare these as local variables.

Problem 15: Climate

Wind chill is a somewhat vague phenomenon associated with the perception of temperature drop that a person feels. There does not exist any standard for wind chill, but in the U.S., the wind chill index T_{wc} is:

$$T_{\text{wc}} = 35.74 + 0.6215T_a - 35.75v^{0.16} + 0.4275T_a v^{0.16} \tag{44}$$

For example, if the temperature is $2\,°\mathrm{C}$ and wind velocity is $5\,\mathrm{mph}$, then the wind chill is $-9\,°\mathrm{C}$.

Deliverables Problem 15

- Complete the function in file hw2.py
- You can create some more solutions to verify your impleImentation is correct.
- Use the math floor function on the value.

Problem 16: Approximation

Most of the time we cannot easily calculate a value, *e.g.*, too big, too small, too complicated. In computing we have learned to guess (or what we call *approximate*). You're no doubt familiar with factorial $n!$, but did you know it was known in the 18th century? Without computers, calculating $n!$ for even what we would consider tiny values was quite difficult. An eponymous approximation, called the Stirling approximation, is (as a lower bound):

$$n! \approx \sqrt{2\pi n}(\frac{n}{e})^n \tag{45}$$

While this might actually look more difficult it isn't. The following code:

```
n0 = 10
print(math.factorial(n0),fact_est(n0))
n0 = n0 * 10
print(math.factorial(n0),fact_est(n0))
```

Produces:

```
3628800 3598695

93326215443944152681699238856266700490715968264381621
46859296389521759999322991560894146397615651828625369
7920827223758251185210916864000000000000000000000000

93248476252694195831713952173205863434091892055096263
58057629519885644096343253465550010174734371150486577
9454898420101234457113230028869790337578478524694528
```

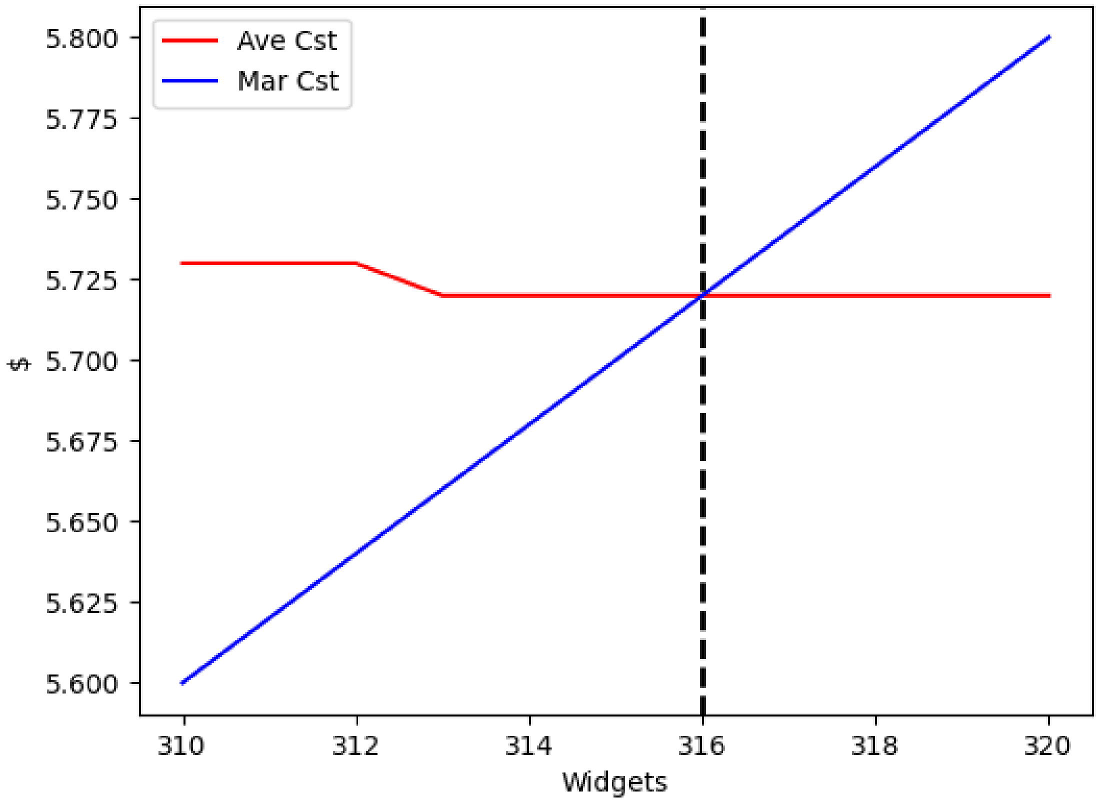
When you run this, it'll look different–because the numbers have so many digits, the values don't easily lend themselves to print.

Deliverables Problem 16

- Complete the function in file hw2.py
- You can create some more solutions to verify your implelmentation is correct.
- Return the approximation as an integer (cast it as an int).

Business for those interested...

Continuing with the cost function, we can use the models to help us make business decisions. How many widgets should we make? The marginal cost describes how cost changes producing additional units. Using both marginal and average cost, we can find the optimal number of widgets to produce by setting them equal and solving for the number of widgets. The marginal cost is the derivative of cost. We'll learn how to approximate this function later in the course. For now, we'll use straight forward calculus. We can see a plot of our answer first:



$$
\begin{aligned}
& \text{marginal cost} \\
\frac{\mathrm{d}C}{\mathrm{d}q} &= 0.02q - 0.6 && (46) \\
& \text{setting both equal} \\
\frac{\mathrm{d}C}{\mathrm{d}q} &= A(q) && (47) \\
0 &= q(0.02q - 0.6) - (0.01q^2 - 0.6q + 1000) && (48) \\
&= 0.02q^2 - 0.6q - 0.01q^2 + 0.6q - 1000 && (49) \\
&= 0.01q^2 - 1000 && (50) \\
q^2 &= 100000 && (51) \\
q &= \lfloor \sqrt{100000} \rfloor = 316 && (52) \\
& && (53)
\end{aligned}
$$

We can confirm this:

$$
\begin{aligned}
C(316) &= \$5.72 && (54) \\
\frac{\mathrm{d}C}{\mathrm{d}q}(316) &= \$5.72 && (55)
\end{aligned}
$$

Here's the code that plotted the values:

```python
import matplotlib.pyplot as plt
import numpy as np

# number of widgets
q = np.arange(310,321)

# Cost
def C(q):
    a,b,c = 0.01,-0.6,1000
    ans = a*(q**2) + b*(q) + c
    return ans

# Average Cost
def A(q):
    return C(q)/q

# Marginal Cost
def marginal_C(q):
    a,b = 0.01,-0.6
    ans = 2*a*(q) + b
    return ans

plt.plot(q,np.round(A(q),2),color='red')
plt.plot(q,np.round(marginal_C(q),2),color='blue')
plt.axvline(x=316,color='black',linestyle = '--',linewidth=2)
plt.legend(["Ave Cst", "Mar Cst"])
plt.xlabel("Widgets")
plt.ylabel("$")
plt.show()
```

We'll learn about all of this in subsequent lectures, but the `numpy` module helps with creating information that is a series of values. `matplotlib` is the most popular visualization module allowing us to create informative graphics of data. They are usually paired together.

Keyboard Quickies

To uncomment multiple lines (Windows):

1. select the lines to be commented
2. Ctrl+k+u

To comment multiple lines (Windows):

1. select the lines to be commented
2. Ctrl+/

To comment or uncomment multiple lines (Mac):

1. select the lines to be commented
2. cmd + /

To undo (Windows):

- Ctrl+Z

To undo (Mac)

- Command+Z