C200 Programming Assignment № 3

Functions & Learning About Homework
Spring 2025

Dr. M.M. Dalkilic

Computer Science

School of Informatics, Computing, and Engineering

Indiana University, Bloomington, IN, USA

February 8, 2025

The purpose of this homework is two-fold: (1) you'll develop your skills in implementing functions, formal and actual parameters, environment and scope (2) how homework will be delivered and submitted. Although the problems can be used for individuals, this course uses paired-programming. Please carefully read about pair-programming at the end of this document.

Please submit the completed HW (the a3.py file)
to Gradescope by **5:00 PM, Friday, Feburary 14**.

Instructions

The course uses an online system for grading Python code, i.e. an *autograder* named Gradescope, which you can find on Canvas on the left side, bar where modules, assignments, etc. are located. The Gradescope autograder has also been linked on the Homework Assignment page itself. Upon login, you can submit HW via 'Upload' button.

(the Gradescope autograder website is `https://www.gradescope.com/`, and can also be accessed using your IU username and password)

Remember that you **must** submit to Gradescope before the HW deadline. **If you don't see the relevant Assignment tab in the Gradescope, please feel free to check back peridiocally. We will also announce on InScribe when we release the assignment on Gradescope.**

Paired-Programming

Homework is generally done with a randomly selected partner. This information can be found under modules on canvas, the page will be called **Student Pairs - Teams for Homework**. **As always, all the work will be with only you and your partner; but *both* of you should contribute.** If your partner does not respond, you must complete the homework on your own. You will not be assigned a new partner–this is infeasible for such a large class. Both students must submit their individual homework, since you are graded individually. Your grade may be different from your partner's grade, since there can be differences in code.

Finding your programming partner: You can find your programming partner by simple text search i.e. use cntrl + F (for windows system) or CMD + F (for MACOS), and type your IU username, press enter. This will highlight the line having your username in the list of programming partners, please, simply look for the name adjacent to your's in the list.

You must complete this before **due date**. You will submit your work by uploading your completed python file to the Gradescope. Please remember that.

- You must submit your individual program even though you have a partner, since you'll be graded individually.
- You cannot discuss homework outside of your partner, use tools like Chat-GPT, copy a solution–a person who cheats cheats himself.
- Struggling is part of any practiced skill–it's how we learn.

Remarks

(1) We do not print or return units–only values. If a function is calculating a percentage, dollar amount, *etc.* the percentage sign % is **not** shown, the dollar sign $ is not shown. For the problem descriptions, we occassionally include them to make interpreting the problem easier.

(2) Often numerical values will be infinite. Python will give you **by default** about 15 decimal places. We'll learn how to shorten these values later. For now, for example we'll write 104.17 for 104.16666666666667. On this HW, you are allowed to use round() function to round the final answer before returning it in the funtion-for example, if your final answer is 6.85457 then it's rounded equivalent is 6.85, and can be obtained by doing **round(6.8547, 2)**. The first parameter to the round() function is the actual value that you want to round, and the second argument is the number of places you want it rounded to. Sometimes, we don't want to round–we simply want the integer portion. Recalling from lecture we can write **int(·)**. For example, int(1.5) = 1, this is also known as *casting*.

(3) Other times we want the ceiling value, the smallest integer greater than or equal to the value. For example, $\lceil 1.5 \rceil = 2$, we can use **math.ceil(·)**, like so **math.ceil(1.5)** = 2. Similarly, we can use **math.floor(·)** to get the floor value, the largest integer less than or equal to the value.

(4) Finally, we can also be exact with values, for example, instead of writing 3.14, we can write **math.pi** to get the value of π. We can also use **math.exp(·)** to get the value of e^x, example: $math.exp(1) \approx 2.7128$. Operations using **math.** are all imported from the math module, we will talk more about this in class, which is already imported in the a3.py file.

Comments about Homework

You will be given the function header/signature, without a body. For each problem, you'll replace the Python keyword **pass** with code. Do not change any function header/signature: meaning

- do not change the function's name, nor the function's arguments/parameters. If you do, the autograder will fail, and you'll receive a zero.

For each of the functions, **make sure to include a docstring that describes the function on line, and the function's signature, i.e. input(s) and output(s), on a separate line.** If you add any extra functions, they must be local to the function you are building; otherwise, the autograder will fail.

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:

$$\text{cone\_volume}(r, h) \quad = \quad \frac{1}{3}\pi r^2 h \tag{1}$$

For example, 2 cm radius and 5 cm height,

$$\text{cone\_volume}(2, 5) \quad \approx \quad 20.94\ cm^3 \tag{2}$$

For 3 cm radius and 7 cm height

$$\text{cone\_volume}(3, 7) \quad \approx \quad 65.97\ cm^3 \tag{3}$$

Deliverables Problem 1

- Complete the *cone_volume* function in the file a3.py.
- Include a docstring with signature for the function.
- 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,
- Example of rounding to different places after decimal are shown below.

```
>>> 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
```

Problem 2: 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 $P(t)$:

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

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

$$P(0) \quad = \quad 31.61\% \tag{5}$$
$$P(3) \quad = \quad 54.02\% \tag{6}$$
$$P(8) \quad = \quad 30.0\% \tag{7}$$

Deliverables Problem 2

- Complete *tv_percent_function* in the file a3.py, using the model above
- Include a docstring with signature for the function.
- Round the answer to 2 decimal places.

Problem 3: 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:

$$I(S) = \lceil 192 \log_2(\frac{S}{762}) - S + 763 \rceil \tag{8}$$

For example,

$$I(100) = 101 \text{ students} \tag{9}$$

$$I(300) = 205 \text{ students} \tag{10}$$

Deliverables Problem 3

- The math module is imported for this problem.
- Complete the *infected_children* function in the file a3.py, using the model above.
- Include a docstring with signature for the function.
- 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 4: Sales Model

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

$$hh(t) = \left\lfloor \frac{532}{1 + 869e^{-1.33t}} \right\rfloor \tag{11}$$

for $0 \leq t \leq 11$ where t is months. The floor function $\lfloor\ \rfloor$ is in the math module named `math.floor()`. The e symbol means exponentiation operation, this is available from `math.exp()` function in math module.

For example,

$$hh(0) = 0 \text{ ducks} \tag{12}$$
$$hh(5) = 250 \text{ ducks} \tag{13}$$
$$hh(10) = 531 \text{ ducks} \tag{14}$$

Deliverables Problem 4

- The math module is already imported in a3.py.
- Complete the *sales_model* function in the file a3.py, using the model above.
- Include a docstring with signature for the function.
- Round your answer to the nearest integer using `math.floor()`

Problem 5: Throwing a Stone

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

$$height(t) = -16t^2 + 64t + 80 \tag{15}$$

The rock will hit the ground after 5 seconds:

$$height(5) = 0 \text{ feet} \tag{16}$$

Deliverables Problem 5

- Complete the *height* function in the file a3.py
- Include a docstring with signature for the function.
- Round the answer to 2 decimal places.

Problem 6: Roots to the Quadratic

Recall that a quadratic is a function:

$$q(x) = ax^2 + bx + c \tag{17}$$

A root is a number that makes the function zero. For example, if

$$q(x) = 2x^2 + 5x - 12 \tag{18}$$

then the two roots are $x = -4$ and $x = 3/2$:

$$q(-4) = 2(-4)^2 + 5(-4) - 12 = 32 - 20 - 12 = 0 \tag{19}$$
$$q(3/2) = 2(3/2)^2 + 5(3/2) - 12 = 2(9/4) + 15/2 - 12 = (24/2) - 12 = 0 \tag{20}$$

Implement a function quad(a,b,c,x) that returns True if x is a root for $ax^2 + bx + c$ and False otherwise. Running the function:

```
print(quad(2,5,-12,-4))
print(quad(2,5,-12,3/2))
print(quad(2,5,-12,1))
```

gives output:

```
True
True
False
```

Deliverables Problem 6

- Complete the *quad* function in the file a3.py
- Include a docstring with signature for the function.

Problem 7: 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:

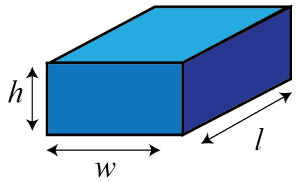
$$R(P, r, n, t) \quad = \quad P \cdot \left[\frac{(1 + \frac{r}{n})^{nt} - 1}{\frac{r}{n}} \right] \tag{21}$$

Deliverables Problem 7

- Complete the R function in the file a3.py
- Include a docstring with signature for the function.
- Round to two decimal places.

Problem 8: Rectangular Solid

Complete the Python function that gives the total surface area S of a rectangular solid with dimensions w, l, h shown here:



The surface area is:

$$S(w, l, h) \quad = \quad (?) \tag{22}$$

Deliverables Problem 8

- Complete the $surface\_area$ function in the file a3.py
- Include a docstring with signature for the function.
- You must create your own solutions for this problem.

Problem 9: 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, however in the U.S., the wind chill index can be calculated by temperature, T_a, and wind speed, v using the following:

$$T_{\text{wc}} \quad = \quad 35.74 + 0.6215T_a - 35.75v^{0.16} + 0.4275T_a v^{0.16} \tag{23}$$

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

Deliverables Problem 9

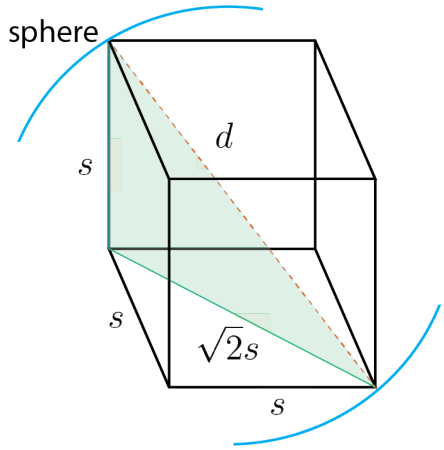
- Complete the T_wc function in file a3.py
- Include a docstring with signature for the function.
- Create some of your own tests to verify your implelmentation is correct.
- Use the math floor function on the value.

Problem 10: Constraints

Given a sphere's volume, v, determine the length of the side of the largest cube inscribed. To help us, we recall the formula for the volume of a sphere:

$$v(r) \quad = \quad \left(\tfrac{4}{3}\right)\pi r^3 \tag{24}$$

We sketch out the cube that includes the sphere's diameter d and sides s of the cube:



Implement two functions `volume_to_radius(v)` that takes the volume of a sphere and gives the radius and `side_max_square(v)` that takes the volume of a sphere and returns the side of the largest cube inscribed. This second function uses the first function. The following code:

```
print(volume_to_radius(v))
print(side_max_square(v))
```

produces:

```
4.0
4.62
```

Deliverables Problem 10

- Complete the functions `volume_to_radius(v)` and `side_max_square(v)` in file a3.py
- Include a docstring with signature for the function.
- Round the return value of `volume_to_radius(v)` and `side_max_square(v)` to 2 decimal places.

Problem 11: Investment Strategy

A simple investment strategy is to fix a dollar amount to purchase a stock; this would be called a *time diversification* strategy. The idea is that, over time, the price should rise following the smaller purchase price. Here's some data:

$ Investement	Market $ Price/Share	Shares
1000	40	250
1000	35	286
1000	34	294
1000	38	263
1000	50	200

that reflects five time periods. To find the average price per share of a sequences of market price per shares $m_1, m_2, \ldots, m_n$, we use the formula:

$$app(m_1, m_2, \ldots, m_n) = \frac{1}{n}\sum_{i=1}^{n} m_i = \frac{1}{n}(m_1 + m_2 + \ldots + m_n) \tag{25}$$

Python has a function `sum()` that takes either a list or tuple and returns the sum:

```
>>> x = [1,2,3,4]
>>> y = (1,2,3,4)
>>> sum(x), sum(y)
(10, 10)
>>> len(x), len(y)
(4, 4)
>>>
```

Recall that `len()` gives the number of members in the list or tuple. Implement a function `app(market)` that takes a list of the market price per shares and returns the tuple `(x,y)` where `x` is the average price per share (rounded to two decimal places) and `y` is the last market price so that we can compare the two. If `x < y` then your strategy has worked. The following code:

```python
market = [40,35,34,38,50]
print(app(market))
```

produces:

```
(39.4, 50)
```

Deliverables Problem 11

- Complete the functions `app(market)` in file a3.py
- Include a docstring with signature for the function.
- Round the return x value of `app(market)` to 2 decimal places.

Communication

When connecting with a new person, you should introduce yourself, give a short message describing the purpose, then salutation. Please use and adapt this format when connecting to a new partner.

```
Dear Student,

My name is X Y and I'm a student in your C200,
but a different section.  It looks like we're
partners this week.  I'm generally free on
Thursday, Saturday, Sunday.  I'm hoping we
can knock most of this out on our first meeting.
Luddy is a great place if we can snag a
conference room.

Take care,

Another Student
```

Format for InScribe post

```
Hi Instructor/ TA <Whom ever they want to
ask question/ Share something>,

If you are posting about lecture/Notes
My name is Student-A. I am in Dr. (mention faculty instructor name)
lecture section. This is regarding ....

If you are posting about labs
My name is Student-A. I am in the lab section that meets on
DAY, TIME of your lab and NAME OF THE TA.
This is regarding ....

Thanks and regards,
<Name>
<User name>
```