C200 Programming Assignment № 1

Functions & Learning About Homework
Fall 2022

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

Indiana University, Bloomington, IN, USA

January 28, 2023

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 below.

Instructions

The course uses an online system for grading the Python code. The system can be accessed at `c200.luddy.indiana.edu`, and you should login via your official IU username and password. Upon login, you can submit HW via 'Create Submission' button. Remember that you **must** push the HW files to GitHub and also submit to the Autograder before the HW deadline.

Paired-Programming

Homework is generally done with a randomly selected partner. In the last section, some information about contacting and communicating with this likely new student will be facilitated. **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.

You must complete this before **Friday, February 03 2023 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. Please remember that.

- You will **not** turn anything in on canvas.
- You do **not manually upload files** to your repository using GitHub's "Upload files" tool.

- You must submit your individual program even though you have a partner, since you'll be graded individually.
- Your github will accept pushes at *any* time. Once the deadline for homework is passed, if you push again, you'll rewrite over the correct time and the solution will be past the deadline and not graded. If your timestamp is 11:01PM or later, the homework will not be graded under almost any circumstance. So do not wait until 10:59PM to commit and push your changes. Pushing with only a few minutes left is **not** a valid reason to be graded.

A remark: 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.

Comments about Homework

You will be given the function signature without a body. For each problem, you'll replace the Python keyword **pass** with code. Do not change any signature. If you do, the autograder will fail and you'll receive a zero. If you add any extra functions, they must be local to the function you are building; otherwise, the autograder will fail. This is a very short homework to help you get started with understanding the process.

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

Some of these problems were taken or inspired by the excellent introductory *Applied Calculus* by Tan, 2005.

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 a1.py.
- You must use math.pi from the math module.
- Round the answer to 2 decimal places. You are allowed to use the python **round()** function to do that.

Problem 2: Oxygen Content of a Pond

The oxygen content t days after the 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 a1.py.
- Round the answer to 2 decimal places. You are allowed to use the python **round()** function to do that.

Problem 3: TV Viewing Patterns

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

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

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

$$P(0) = 31.6074 \tag{8}$$
$$P(3) \approx 54.0225 \tag{9}$$
$$P(8) \approx 29.99999999999982 \tag{10}$$

Deliverables Problem 3

- Complete tv percent function in the file a1.py.
- Round the answer to 2 decimal places. You are allowed to use the python **round()** function to do that.

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 a1.py.
- Round the answer to 2 decimal places. You are allowed to use the python **round()** function to do that.

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:

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

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

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

(17)

Deliverables Problem 5

- Complete Cowling's rule function in the file a1.py.
- Round the answer to 2 decimal places. You are allowed to use the python **round()** function to do that.

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:

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

Deliverables Problem 6

- The math module is imported for this problem.
- Complete the I function in the file a1.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.
- Use Python's **math.ceil()** function to round up your answer to the nearest integer. **ceil(7.1) = 8**, **math.ceil(7.8) = 8**.

Problem 7: Average Cost

Let $C(q)$ be the cost of producing q items. The **average cost** $A(q)$ is given by

$$A(q) = \frac{C(q)}{q} \tag{19}$$

A company's cost function is (in $) given by

$$C(q) = 0.01q^3 - 0.6q^2 + 13q + 1000 \tag{20}$$

Deliverables Problem 7

- Complete both functions C, A in the file a1.py
- For this problem, you'll need to make your own input/outputs

Problem 8: Sales Model

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

$$hh(t) = \left\lfloor \frac{532}{1 + 869e^{-1.33t}} \right\rfloor \tag{21}$$

for $0 \leq t \leq 11$ where t is months. Remember, we looked at the floor function $\lfloor \rfloor$ in the math module named floor. For example,

$$hh(0) = 0 \tag{22}$$
$$hh(5) = 250 \tag{23}$$
$$hh(10) = 531 \tag{24}$$

Deliverables Problem 8

- The math module is imported for this problem.
- Complete the hh function in the file a1.py.
- Make sure that you round down your answer to the nearest integer by using floor function.

Problem 9: Throwing a Stone

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

$$height(t) = -16t^2 + 64t + 80 \tag{25}$$

Deliverables Problem 9

- Complete the $height$ function in the file a1.py
- Round the answer to 2 decimal places. You are allowed to use the python **round()** function to do that.

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:

$$time(t) = \frac{0.44t^4 + 700}{0.1t^4 + 7} \tag{26}$$

for $(0 \leq t \leq 24)$

Deliverables Problem 10

- Complete the $time$ function in the file a1.py.
- For this problem, you'll need to make your own input/outputs.
- Round the answer to 2 decimal places. You are allowed to use the python **round()** function to do that.

Problem 11: Roots to the Quadratic

Recall that a quadratic is a function:

$$q(x) = ax^2 + bx + c \tag{27}$$

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

$$q(x) = 2x^2 + 5x - 12 \tag{28}$$

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

$$q(-4) = 2(-4)^2 + 5(-4) - 12 = 32 - 20 - 12 = 0 \tag{29}$$

$$q(3/2) = 2(3/2)^2 + 5(3/2) - 12 = 2(9/4) + 15/2 - 12 = (24/2) - 12 = 0 \tag{30}$$

On the other hand, 1 is not. Implement a function quad(a,b,c,v) that returns True if v is a root for $ax^2 + bx + c$ and False otherwise. Running the function:

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

gives output:

```
1 True
2 True
3 False
```

Deliverables Problem 11

- Complete the $quad$ function in the file a1.py

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. The student pairs are given on the next page.

```
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>
```

Student Pairs

aaberma@iu.edu, hmerrit@iu.edu
actonm@iu.edu, jmom@iu.edu
adagar@iu.edu, nireilly@iu.edu
brakin@iu.edu, nmonberg@iu.edu
sakinolu@iu.edu, ogift@iu.edu
moalnass@iu.edu, jacklian@iu.edu
anderblm@iu.edu, lulacayo@iu.edu
jaybaity@iu.edu, joshbrin@iu.edu
nbakken@iu.edu, cartmull@iu.edu
chnbalta@iu.edu, joracobb@iu.edu
nokebark@iu.edu, moesan@iu.edu
nwbarret@iu.edu, yz145@iu.edu
sbehman@iu.edu, jackssar@iu.edu
jadbenav@iu.edu, esmmcder@iu.edu
arjbhar@iu.edu, edshipp@iu.edu
dombish@iu.edu, ornash@iu.edu
gavbish@iu.edu, roelreye@iu.edu
sebisson@iu.edu, jonhick@iu.edu
aibitner@iu.edu, ekkumar@iu.edu
abolad@iu.edu, jcpilche@iu.edu
hjbolus@iu.edu, klongfie@iu.edu
jacbooth@iu.edu, parkecar@iu.edu
neybrito@iu.edu, jmissey@iu.edu
nbulgare@iu.edu, sjvaleo@iu.edu
ivycai@iu.edu, sijatto@iu.edu
bcarl@iu.edu, jl263@iu.edu
ecastano@iu.edu, fshamrin@iu.edu
cheng47@iu.edu, bashih@iu.edu
kbchiu@iu.edu, zisun@iu.edu
hc51@iu.edu, sl92@iu.edu
sgcolett@iu.edu, swcolson@iu.edu
joecool@iu.edu, zhaozhe@iu.edu
jacosby@iu.edu, kynogree@iu.edu
quecox@iu.edu, schwani@iu.edu
jacuau@iu.edu, sihamza@iu.edu
aadidogr@iu.edu, clschel@iu.edu
siqidong@iu.edu, lgflynn@iu.edu
wdoub@iu.edu, cmarcucc@iu.edu

dud@iu.edu, johwarre@iu.edu
marbelli@iu.edu, aketcha@iu.edu
abdufall@iu.edu, jvalleci@iu.edu
stfashir@iu.edu, ttieu@iu.edu
sfawaz@iu.edu, webejack@iu.edu
jfearri@iu.edu, tifhuang@iu.edu
tyfeldm@iu.edu, swa5@iu.edu
tflaa@iu.edu, rmatejcz@iu.edu
megofolz@iu.edu, dloutfi@iu.edu
bgabbert@iu.edu, ss126@iu.edu
mgambett@iu.edu, agoldswo@iu.edu
mdgamble@iu.edu, lsherbst@iu.edu
jegillar@iu.edu, chrinayl@iu.edu
egillig@iu.edu, greymonr@iu.edu
mgorals@iu.edu, jonvance@iu.edu
davgourl@iu.edu, jyamarti@iu.edu
qugriff@iu.edu, ghyatt@iu.edu
aguarda@iu.edu, patel88@iu.edu
cahilber@iu.edu, jarymeln@iu.edu
achimeba@iu.edu, imalhan@iu.edu
anthoang@iu.edu, aalesh@iu.edu
qhodgman@iu.edu, tshore@iu.edu
hollidaa@iu.edu, cjromine@iu.edu
rythudso@iu.edu, eddykim@iu.edu
huntbri@iu.edu, danwils@iu.edu
bradhutc@iu.edu, vilokale@iu.edu
ajarillo@iu.edu, cartwolf@iu.edu
ljianghe@iu.edu, svaidhy@iu.edu
coldjone@iu.edu, shethsu@iu.edu
sjkallub@iu.edu, dannwint@iu.edu
dilkang@iu.edu, mvanworm@iu.edu
pkasarla@iu.edu, alescarb@iu.edu
katzjor@iu.edu, kvanever@iu.edu
arikilli@iu.edu, bpoddut@iu.edu
dk80@iu.edu, wallachl@iu.edu
nklos@iu.edu, akundur@iu.edu
arykota@iu.edu, kalomart@iu.edu
delkumar@iu.edu, sabola@iu.edu
aladkhan@iu.edu, yijwei@iu.edu
jolawre@iu.edu, mw154@iu.edu

jl335@iu.edu, caldsmit@iu.edu
georliu@iu.edu, wuyul@iu.edu
lopeal@iu.edu, dwinger@iu.edu
jasoluca@iu.edu, ajneel@iu.edu
rpmahesh@iu.edu, yasmpate@iu.edu
lmamidip@iu.edu, pvinod@iu.edu
mmandiwa@iu.edu, rair@iu.edu
crfmarti@iu.edu, btpfeil@iu.edu
imaychru@iu.edu, grschenc@iu.edu
nkmeade@iu.edu, lenonti@iu.edu
omilden@iu.edu, leplata@iu.edu
fimitch@iu.edu, criekki@iu.edu
almoelle@iu.edu, avpeda@iu.edu
isarmoss@iu.edu, macsvobo@iu.edu
masmuell@iu.edu, sowmo@iu.edu
cmulgrew@iu.edu, hvelidi@iu.edu
hnichin@iu.edu, davthorn@iu.edu
chnico@iu.edu, snuthala@iu.edu
parisbel@iu.edu, saseiber@iu.edu
arnpate@iu.edu, sousingh@iu.edu
dpepping@iu.edu, marystre@iu.edu
elyperry@iu.edu, bensokol@iu.edu
caitreye@iu.edu, mszczas@iu.edu
psaggar@iu.edu, mostrodt@iu.edu
crsaylor@iu.edu, dy11@iu.edu
peschulz@iu.edu, aw149@iu.edu
alexschu@iu.edu, gsprinkl@iu.edu
shahneh@iu.edu, sjxavier@iu.edu
js153@iu.edu, vthakka@iu.edu
voram@iu.edu, jwescott@iu.edu
trewoo@iu.edu, jorzhang@iu.edu