PET328 2022-2023 Exam Solution

August 4, 2023

[1]:
```
print('... to the only wise God')
```

```
… to the only wise God
```

1 Context

Congratulations on your recent placement as a Reservoir Engineering Intern! The Senior Reservoir Advisor (SRA) and the Lead Software Developer (LSD) are your training mentors, and have tasked you with the following requests. A custom Python module, peteng.py being built and used by the TTOWG Asset Team is herewith attached.

2 Question 1

2.1 a.

Question

Given that you are developing a script to be used by members of the team, write a Python statement to request an ID string from a user. Why would it be unnecessary to convert the user's response to a string? (2 marks)

[ ]:
```
id = input("What is your user ID?\n")       # 1 mark
```

It would be unneccasry to convert the user's response to a string because Function *input* stores the response as a string, by defualt. ***1 mark***

2.2 b.

Question

Write a Python statement to make Module peteng available in your script, in anticipation of needing most of the functions therein. Take note that it is desirable to avoid prefixing with file name. (2 marks)

[ ]:
```
from peteng import*       # 2 marks
```

2.3 c.

Question

Extend the capability of Module peteng by including Function archie_sw according to the equation: $s_w = \sqrt[n]{\frac{aR_w}{\phi^m R_t}}$

You may pass all parameters as function arguments. Let the return value be in 4 decimal places. (3½ marks)

[ ]:
```
def archie_sw(n, a, rw, poro, m, rt):      # 1/2 mark
    sw = ((a*rw)/((poro**m)*rt))**n        # 1/2 mark
    sw = round(sw,4)                       # 1/2 mark
    return sw                              # 1/2 mark
```

2.4 d.

Question

Respond to the following questions from the LSD who is testing your understanding of Module peteng.

i. What is the implication, for users, of defaulting some arguments of gas_density (Line 8)? (2 marks)

ii. What happens if a user calls Function fvf (Line 41) without specifying value for Argument pb? (2 marks)

iii. Create an object to be passed as argument to Function stoiip_2 (Line 72), use arbitrary values. (2 marks)

iv. Why is Function str (Line 118) invoked? (2 marks)

v. What is the essence of Lines 133 and 134? (2 marks)

i. Those defaulted arguments become optional; users can call the function without specifying values for the defaulted arguments. ***2 marks***

ii. If a user did not specify value for pb, Function fvf itself computes pb values internally by calling functions bubble_pressure and sol_gor. The internally-computed value of pb then overides the default value (None) and get used in computation. ***2 marks***

[ ]:
```
# iii.

data_dict = {'area': 40, 'thickness': 15, 'poro': 0.28, 'swi': 0.3, 'boi': 1.2}␣
↪    **2 marks**
```

iv. Function *str* is invoked to convert *block_n_order*, which is a float, to a string; so that it could be concatenated with the String *'Block'* ***2 marks***

v. Lines 133 and 134 are written to initialize *total_stoiip* and *np_list*; so that the two objects can then be populated with values inside the subsequent *'for'* loops. ***2 marks***

3 Question 2

3.1 a.

Question

Give a sequence of GitHub operations you would need to implement in order to have offline access to project files in a repository that the SRA just shared with you. (2 marks)

GitHub Workflow: Fork the repository ——-> Clone the forked repository. ***2 marks***

3.2 b.

Question

Write a Python code to check that a user's input is numeric, and to display "Numeric input required" in case the user's input is non-numeric. (2 marks)

[18]:
```
if not(type(user_inp) == float or type(user_inp) == int):          # 1.5 mark
    print("Numeric input required")      # 0.5 mark
```

```
Numeric input required
```

3.3 c.

Question

Create Function hydrostat_p that would compute drilling mud hydrostatic pressure at a given true vertical depth, D, according to the following algorithm: $HP = 0.52\rho_m D$ if mud densiity, ρ_m is known, or, $HP = \delta_m D$ if pressure gradient, δ_m, is known. (3½ marks)

[5]:
```
def hydrostat_p(d, rho_m = None, delta_m = None):      # 1 mark
    if delta_m is None:      # 0.5 mark
        hp = 0.52*rho_m*d    # 0.5 mark
    else:                    # 0.5 mark
        hp = delta_m*d       # 0.5 mark
    return round(hp, 2)      # 0.5 mark
```

3.4 d.

Question

The SRA requests for a new reservoir volumetric parameter, gross reservoir volume (GRV), based on trapezoidal rule. Using the following algorithm, create Function *grv* to accept *owc*, *top* and *area_list* as arguments and returns *vol*. The terms a_o and a_n refers to the first and the last elements of *area_list* while other elements are denoted as a_1, a_2, ... a_{n-1}. Be aware that n is the number of elements in *area_list*. (10 marks)

$$GRV = h(\frac{1}{2}[a_0 + a_n] + [a_1 + a_2 + ... + a_{n-1}])$$

$$h = \frac{owc - top}{n}$$

[13]:
```
def grv(owc, top, area_list):      # 1 mark
    n = len(area_list)             # 2 marks
    h = (owc - top)/n              # 1 mark
    vol = h*((0.5*(area_list[0] + area_list[n-1])) + sum(area_list[1:n-1]))
                          # 1 mark          1 mark     1 mark       2 marks
    return round(vol, 2)  # 1 mark
```

[ ]:
```

```

4 Question 3

4.1 a.

Question

Given that 0.26 and 850 have been assigned to 'poro' and 'perm', respectively. Create a list of tuples, with each tuple holding a pair of the variable name and value. Convert the list to a dictionary object. (2 marks).

[3]:
```
my_list = [('poro', 0.26),('perm', 850)]      # 1 mark
my_dict = dict(my_list)       # 1 mark
```

4.2 b.

Question

Re-construct the following statement in a way that the conditions are based on the "less than or equal to" Boolean operator:

(2 marks)

[ ]:
```
if co2_comp > 0.12 or n2_comp > 0.03 or h2s_comp > 0
```

[5]:
```
not(co2_comp <= 0.12 and n2_comp <= 0.03 and h2s_comp <= 0)        # 2 marks
```

4.3 c.

Question

Well *CH_23* is to be drilled in successive depth segments; cost of drilling a segment is: $cost = B + C_r(t + T)$. Construct a loop to compute the cost per segment and to obtain (by summation) the aggregate/cumulative cost, until the total drilled depth exceeds 10,000 ft. You may denote the depth per segment as *delta_D*. (3½ marks)

[9]:
```
aggreg_cost = 0      # 0.5 mark
total_depth = 0      # 0.5 mark

while total_depth <= 10000:     # 1 mark
    cost = B + cr*(t + T)       # 0.5 mark
    aggreg_cost = aggreg_cost + cost    # 0.5 mark
```

```
    total_depth = total_depth + delta_D      # 0.5 mark
```

4.4 d.

Question

Write a script to implement the following workflow, in response to a request from the Senior Reservoir Advisor:

- Make Function *stoiip_discretized* from Module peteng (Line 77) available for call. (2 marks)
- Create a Function *trimmed_sum* that receives *inp_list* and *threshold_val* (a float) and returns the sum of all elements of inp_list whose values are greater than threshold_val. (4 marks)
- Call Function *stoiip_discretized* and pass the *stoiip_list* component of its output as argument to a call of Function *trimmed_sum*. (4 marks)

[ ]:
```
# Making Function stoiip_discretized available
from peteng import stoiip_discretized      # 2 marks

# Defining Function trimmed_sum
def trimmed_sum(inp_list, threshold_val):      # 1 mark
    the_sum = 0         # 0.5 mark
    for value in inp_list:      # 0.5 mark
        if value > threshold_val:       # 0.5 mark
            the_sum = the_sum + value    # 1 mark
    return the_sum        # 0.5 mark

# Calling Function stoiip_discretized
(stoiip_value, the_list) = stoiip_discretized(100, 100, 40, 2, 1, 1.12, [0.25,␣
 ↪0.31], [0.28, 0.30])
#  1 mark                          1 mark                                        0.5␣
 ↪mark        0.5 mark

# Passing the stoiip_list component to trimmed_sum
trimmed_sum(the_list, the_threshold)      # 1 mark
```

5 Question 4

5.1 a.

Question

What do you think would go wrong if Line L4a below is to be replaced with Line L4b, as per Function gas-density (Line 8 of peteng)?

[ ]:
```
gas_density(gravity = 0.786, pressure = 14.7, temperature = 520, z = 1)      # -␣
 ↪- - - L4a
gas_density(0.786, 520)      # - - -- - - L4b
```

The function would erroneously assign the value 520 to the pressure argument, instead of assigning it to temperature.

5.2 b.

If you are to replace the list in Lines 87 and 96 of *peteng* with a tuple object, how would you implement the *.append* effect, considering that tuples are immutable and do not have the *.append* method? (2 marks)

[4]:
```
stoiip_tuple = stoiip_tuple + (block_stoiip,)
```

5.3 c.

Create Function *darcy_rate* according to the equation $q = C_f \frac{KA\Delta P}{\mu L}$. You may pass all parameters as function arguments. Parameter C_f is the conversion factor and is often (but sometimes not) equal to 0.001127. Let the return value be in 2 decimal places. (3½ marks)

[ ]:
```
def darcy_rate(perm, area, del_p, visc, length, c_f = 0.001127):      # 1 mark
    rate = (c_f*perm*area*del_p)/(visc*length)    # 1 mark
    rate = round(rate, 2)    # 1 mark
    return rate    # 0.5 mark
```

5.4 d.

For a given discretized reservoir model, the permeability values of some gridblocks are so low that the SRA would like gridblocks with permeability lower than a given cut-off value to be classified as '*inactive*' while others are to be classified as '*active*'. Create Function *block_classifier* that receives *perm_list* and *cut_off* (a float) and returns a dictionary of *block_ID* (as keys) and string '*inactive*' or '*active*' (as values). Keys *block_ID* should contain string '*Block*' and the natural ordering of the blocks (e.g. '*Block1*') (10 marks)

[1]:
```
def block_classifier (perm_list, cut_off, nx, ny):
    classification_dict = {}
    for j in range(1,ny+1):
        for i in range(1,nx+1):
            block_n_order = (nx*(j-1))+i
            block_ID = 'Block'+str(block_n_order)
            if perm_list[block_n_order - 1] < cut_off:
                classification_dict[block_ID] = 'inactive'
            else:
                classification_dict[block_ID] = 'active'
    return classification_dict
```

[ ]:
```

```