COVENANT UNIVERSITY
CANAANLAND, KM 10, IDIROKO ROAD
P.M.B 1023, OTA, OGUN STATE, NIGERIA

TITLE OF EXAMINATION: B. Eng. EXAMINATION
COLLEGE: ENGINEERING
DEPARTMENT: PETROLEUM ENGINEERING
SESSION: 2021/2022 **SEMESTER:** OMEGA
COURSE CODE: PET328 **CREDIT UNIT:** 2
COURSE TITLE: COMPUTER APPLICATIONS IN PETROLEUM ENGINEERING
INSTRUCTIONS: Attempt to answer ALL questions
Sharing of any material whatsoever is highly prohibited (Article 2.6.12 of CU Exam Manual). **TIME:** 2 hours

---

The Superintendent Engineer (SE) of TOWG Asset Team has initiated a project to automate several routine oilfield computations needed in his team. He recently added you to the automation project group. The following questions are the set of tasks he wanted you to perform. A Python module for this project, *peteng.py*, is made available for this exam.

Question 1: [17½ marks]

a. Justify the need for Funtion `float` in Line 85 of Module `peteng` **[2 marks]**
b. Construct a Boolean expression that would evaluate to `True` when `poro` is greater than `poro_cutoff` and `perm` is greater than `perm_cutoff`. **[2 marks]**
c. Function `stoiip`, as defined in Lines 49- 51 of Module `peteng`, accepts Argument `area` in acre units. Sometimes, it is preferable to have reservoir dimensions (`length_x` and `length_y`) in feet as the function arguments, and let the function internally compute `area` as $length\_x \times length\_y$. Re-create the function so as to permit users to input either `area` and skip dimensions or vice-versa. Note: 1 acre = 43560 ft^2. **[3½ marks]**
d. Demonstrate your understanding of Module penteng by writing lines to implement the following calls:
   i. Call Function `gas_density` (Line 3) with `gravity`, `pressure` and `z` set to 0.79, 2000psia and 0.895, respectively. **[2 marks]**
   ii. Call Function `fvf` (of Line 30) with `pressure`, `temperature`, `gas_gravity`, `oil_gravity`, `pb` and `co` set to 4000psi, 220^{0}F, 0.786, 0.8217, 2609psi and 14.13×10^{-6}psi^{-1}, respectively. **[2 marks]**
   iii. Call Function `fvf` (of Line 30) with `pressure`, `temperature`, `gas_gravity`, `oil_gravity` and `pb` set to 2100psi, 220^{0}F, 0.786, 0.8217, 2609psi, respectively. **[2 marks]**
   iv. How are the calls in Questions 1.d.ii - iii possible without setting values for some arguments? **[2 marks]**
   v. Create a dictionary object to hold 50 acres, 27ft, 0.23, 0.28 and 1.19RB/STB as values of `area`, `thickness`, `poro`, `swi` and `boi`, respectively. Pass the dictionary object as argument in a call to Function `stoiip_2` (of Line 55 - 57).**[2 marks]**

Question 2: [17½ marks]

a. Which line(s) in `peteng` would be skipped if Function `fvf` is called without specifying Argument `rs`? **[2 marks]**
b. Which of the `for` loops in Lines 70 - 71 iterates through columns of a discretized reservoir model? **[2 marks]**
c. You observed the challenge of project files transfer between the SE and members of his team, and you are determined to introduce distributed version control (DVC) to the team. Present a simple workflow chart to explain the processes by which the SE and team members can collaborate on such files. **[3½ marks]**
d. It is desirable to compute other volumetric reservoir potentials (bulk volume (bv) pore volume (pv), hydrocarbon pore volume (hcpv)), in addition to STOIIP currently being computed by Function stoiip_discretized_2 of Line 60 - 79. Using these lines as the basis, create Function `volumetrics` to compute and return `total_bv`, `total_pv`, `total_hcpv` and `total_stoiip`, as dictionary items, for a 2-D discretized reservoir, according to the following equations:

$$block\_bv = delta\_x \times delta\_y \times h$$
$$block\_pv = block\_bv \times poro$$
$$block\_hcpv = block\_pv \times (1 - sw)$$
$$block\_stoiip = \frac{block\_hcpv}{5.615 \times boi}$$

Note: the function is NOT required to return block-level volumetrics such as `block_stoiip`. **[10 marks]**

Question 3: [17½ marks]

a. Write a line to be inserted in Module `peteng` to ensure Function `gas_density` (Line 3) always display its output on screen. **[2 marks]**

b. Which function makes the import statement in Line 42 of Module `peteng` necessary?**[2 marks]**

c. Given two lists, `thickness_vals` and `perm_vals`, respectively containing values of thickness and permeability of some reservoir zones. Construct a loop to iterate through these lists and compute the average permeability for a linear series flow system thus:

$$avg\_perm = \frac{\sum_i^N h_i}{\sum_i^N \frac{h_i}{k_i}}$$

where N is number of zones, k_i and h_i are permeability and thickness of Zone i, respectively **[3½ marks]**

d. Create Function `formation_compr` capable of using any of the following correlations to compute and return the value of formation compressibility for a reservoir.

Hall Correlation: $c_f = \left(\frac{1.782}{\phi^{0.438}}\right) \times 10^{-6}$; Newman Correlation: $c_f = \frac{a}{(1+bc\phi)^{\frac{1}{b}}}$

where:

$a = 97.32 \times 10^{-6}, b = 0.699993, c = 79.8181$ - for sandstone reservoirs

$a = 0.8535, b = 1.075, c = 2.202 \times 10^6$ - for limestone reservoirs

In addition to Argument `poro`, the function should have Arguments `corr_type` and `formation_type` to enable users specify the preferred correlation and the applicable formation type, respectively. **[10 marks]**

Question 4: [17½ marks]

a. What is the purpose of Line 68 of Module `peteng`? **[2 marks]**

b. Why is Line 38 of Module `peteng` outside the conditional structure of Lines 39 - 45? **[2 marks]**

c. Currently, Module `peteng` contains an implementation of Baye's Theorem for facies classification, at Lines 83 - 102. Based on these lines, create a function named `facie_bayes`. Let the function return both the facie classification (`facie_class`) and the associated probability (`prob`).Take note that:

$$cond\_prob\_shale = 1 - cond\_prob\_dolomite$$

[3½ marks]

d. Create Function `pwf_updator`, with Arguments `gwi`, `visc`, `b`, `pi_list`, `pwfc` and `qsp`, to implement the following algorithm needed for the constant-rate regime of a reservoir simulator script:

   i. initialize counter `j` as 1, `pwf` as $pwfc + 1$, and `pwf_tuple` as an empty tuple

   ii. perform the following in a while loop as long as `pwf` is greater than `pwfc`

      a. set `qsc` to be equal to `qsp`

      b. set `pi` to be equal to the jth element in `pi_list`

      c. update `j` as $j = j + 1$; `pwf` as $pwf = p_i + \frac{qsc \times visc \times b}{gwi}$; and `pwf_tuple` with the latest `pwf`.

   iii. At loop termination, print 'End of constant rate regime!' and return `pwf_tuple`.

```
print("Best Wishes!!!")
```