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: 2020/2021 **SEMESTER:** OMEGA
COURSE CODE: PET328 **CREDIT UNIT:** 2
COURSE TITLE: COMPUTER APPLICATIONS IN PETROLEUM ENGINEERING
INSTRUCTIONS: Attempt to answer ALL questions
 TIME: 2 hours

You have just secured a placement as a Reservoir Engineering Intern, and have been deployed to an oilfield asset team. 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 team is herewith attached.

Question 1: [17½ marks]

a. What chain of GitHub workflow would you implement in order to inform the Lead Software Developer of edits you have made to the project code files? **[2 marks]**

b. The following code snippet performs some corrections to *gas_gravity* if some impurity thresholds are violated. Re-write this snippet to communicate the message 'Correction not needed' if none of threshold is violated.

```
if co2_comp > 0.12 or n2_comp > 0.03 or h2s_comp > 0:
    gas_gravity = some_computations
    print('The corrected gas gravity is', gas_gravity)
```

[2 marks]

c. Write simple lines of codes to create Function *volumetrics* that accepts arguments area (ft), thickness (ft) and porosity (decimal fraction); and returns both the bulk volume (BV) and pore volume (PV) of a reservoir. How would you call the function in such a manner as to unbundle the return values? **[3½ marks]**

d. Demonstrate your understanding of Module penteng by answering the following questions:

i. Why is 1 added to *ny* in Line 78? **[2 marks]**
ii. There is a bug in Line 102; identify it and fix it? **[2 marks]**
iii. Why is 1 subtracted from *block_n_order* in Line 81? **[2 marks]**
iv. Which other List method could have been used in Line 84. **[2 marks]**
v. Write a line to replace Line 84, if a tuple had been initialized in Line 76. **[2 marks]**

Question 2: [17½ marks]

a. What chain of GitHub workflow would you implement in order to update your offline copy of project files with contributions from your collaborators? **[2 marks]**

b. What is the purpose of Line 76 of Module *peteng*? **[2 marks]**

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]**

d. Create Function *mat_bal* to execute the MBE $\left(N_p = \frac{NB_{oi}c_e(P_i - P_{now})}{B_o}\right)$ for a discretized reservoir model. Take note that the Bo term is to be updated thus: $B_o = B_{ob}[1 - c_o(P_{now} - P_b)]$. The grid of pressure values

should be passed to the function as *pnow_list*; with the list index corresponding to the natural ordering of the grid blocks. The function should return *np_list* containing all *block_np* values, with elements arranged in natural ordering of the block, and *total_np*. You may pass all parameters of these equations (except Bo) as function arguments. **[10 marks]**

Question 3: [17½ marks]

a. Following the principle of sequential execution, write a line of code that you think should precede the following:

```
for i in range(1,nx+1):
        block_stoiip = some_computations
        stoiip_list.append(block_stoiip)
```

[2 marks]

b. In Line 37, both Arguments *co* and *rs* are optional in the context of users' specification. Which one of them is NOT optional in the context of function's workings? Explain, please. **[2 marks]**

c. Oil reservoir material balance simulators are sometimes written only for undersaturated conditions. In such cases, the program is written to run as long as average reservoir pressure is still greater than bubble-point pressure. Write a 'while' loop (not the full function) that executes $N_p = \frac{NB_{oi}c_e(P_i - P_{now})}{B_o}$ as long as *average_pressure* is greater than *bubble_pressure*, and terminates (with message 'Bubble-point attained'), otherwise. **[3½ marks]**

d. For a given discretized reservoir model, the permeability values of some gridblocks are so low that the Senior Reservoir Advisor 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]**

Question 4: [17½ marks]

a. Construct a logical statement that would evaluate to True as long as any of the following condition is fulfilled:

- Condition 1: *current_pressure* is below *bubble_pressure*
- Condition 2: *flow_rate* is lower than *economic_limit*

[2 marks]

b. If temperature T = 600°R is intended; point out the error in the following call to Function *gas_density* defined in Line 7 of Module peteng: `gas_density(0.786, 600)`. Fix the bug, please.

c. Consider *stoiip_dict* being one of the values returned by Function *stoiip_discretized_2* defined in Line 89 of Module *peteng*. Currently, when *stoiip_dict* is printed, all items are displayed on a single line; this is clumsy! Write a 'for' loop to print each item of *stoiip_dict* on a separate line. **[3½ marks]**

d. Write a program to implement the following workflow, being some initial efforts at creating Sub-module gas_props in Module peteng:

- Create Function *appar_molweight* that receives *molfrac_list* and *molweight_list* respectively containing the mole fractions and molecular weight of some components in a natural gas mixture. Let the function return the apparent molecular weight of the mixture, computed thus: $M_{app} = \sum_i^N y_i M_i$; N is the number of components in the mix.
- Create Function *gas_density* that receives *P, T, R* (defaulted to 10.73), *molfrac_list*, *molweight_list* and *Mapp*; defaulting the last three arguments to None. Let the function return the density of the gas mixture given as: $\rho_g = \frac{PM_g}{RT}$. Let Function gas_density call Function *appar_molweight* if necessary.