IC152: Assignment 11 OOP, Stacks and Queues

- **<u>This is a group assignment.</u>**
- **<u>Write algo with pen paper before code. 1 algo per group is fine.</u>**

Problem 1: Object Oriented Programming (OOP), Stacks and Queues

a. Create a class IC152 which captures the name, roll no, attendance, marks and pass_flag (True/False) obtained by a student for IC152 instance attributes, and attendance_percentage_criteria as class attribute.

- The code takes as inputs: the name and roll no of the student through constructor.

- Before you create an object, make sure that the object is of proper form by checking the following things

  • Check the roll no to be of the form B24*** or IM24*** where *** represents a string with a three digit number.

  • Also set the instance attributes “marks”, “attendance” and “pass_flag” to be 0, 0 and False respectively by default.

  • Set the class variable as attendance_percentage_criteria to be 80 by default

- take the inputs name and roll no through constructor and if the input is valid, print “valid” (without quotes). If the input is invalid, print “invalid input” (without quotes). DO NOT TAKE OTHER INPUTS THROUGH CONSTRUCTOR. Accordingly create the input_update_method() for updating the object created with invalid input.

- Write an instance method that takes marks of the student and assigns it to an instance attribute. If the student has secured at least 33 marks, another instance attribute “pass_flag” should be set as False if the student has secured less than 33 marks. If the score is 33 or above the instance attribute “pass_flag” should be set as True.

  • Check the marks to be in the range of 0 to 100 only when called through the instance method.

  • When called through the instance method: The marks should be automatically updated to 0 in case negative marks are entered, and 100 in case more than 100 marks are passed as input to the constructor.

  - Write an instance method to update the “attendance” attribute (e.g. 82 represents 82 %) and update the instance attribute “pass_flag” as False in case attendance is below value present in class attribute attendance_percentage_criteria.

  - Create 3 student objects one with 29 marks and 85 attendance, another with 32 marks and 70 attendance, and another with marks 99 with 80 attendance. Suppose the instructor wants to give +5 marks to all the students due to an error in a question in the end sem. Use instance/class attribute and instance/class method appropriately to correctly update the “marks” and “pass_flag” attribute of the three students.

b. Create a class with a name queue, with queue_list (name of a list) and element as attributes. Write three methods under the queue class for push(), pop(), and peek() operations. The queue is a limited access data structure that follows FIFO (first in first out) rule.
  - The push() method should push its input at the end/top of queue_list.
  - The pop() method should remove the first/lowest element of queue_list, update it in the element attribute and return it as output.
  - The peek() method should just update the first/lowest element of queue_list in the element attribute and return it as output.
  - IMPORTANT: You are NOT allowed to use any of the list methods. You can use loops and indexing.
  - Test your code by giving different examples.

Problem 2:

Given a string of single-digit integers, binary operands and brackets/parentheses, find out if the brackets/ parentheses are balanced.

- In a stack, one can add and remove elements only from the top. They are referred to as push and pop operations. Hence, while performing a pop, the element which is retrieved is the one which was pushed the last. In other words, last-in-first-out.
- A stack can be used for finding out if the parentheses are balanced in the input string, as follows. Each time an opening parenthesis is encountered in the input string, push it on the stack. Each time a closing parenthesis is encountered, pop an element from the stack and check if the opening and closing parentheses are of matching type. If not, then you know that the string is unbalanced. Integers and binary operands may be ignored. If at the end, there are extra opening/closing parenthesis, then again, you know that the string is unbalanced. Return 1 if the string is balanced and 0 otherwise.
- Try following test cases:
  1 ∗ (2 + 3 ∗ [4 − 5]) should return 1.
  (1 + [2 + 3]) should return 1.
  (1 + [2) + 3] should return 0.
  1 ∗ (2 + 3 ∗ [4 − 5}) should return 0.
  1 ∗ (2 − 3 should return 0.