Mock test

Program(studentGrading.py):

```
#create a class to perform various functions
class Grade:
  def __init__(self):
    self.marks = []
    self.result = []
    self.total = 0

  #function to obtain number of students from the user == N students
  def get_marks(self):
    self.number = int( input( 'Enter how many students:'))
    for i in self.number:
      self.marks.append( 'Enter the marks: ')

  #function to calculate the mark and add them in a list
  def calculate_mark(self):
    self.result = []
    if self.marks < 0 and self.marks >100:
      if (self.marks >= 40):
        self.result.append('Pass')
      else:
        self.result.append('Fail')
    else:
      self.result.append('Invalid Marks')
    return self.result
```

```
#function to calculate the total marks
def total_marks(self):
   self.total = 0
   for i  in self.marks:
      total = total + i
      print('Total marks is: ', total)
   return self.total

#function to calculate the average marks
def average_marks(self):
   self.average = self.total / self.number
   print('The average marks is: ', self.average)
   return self.average

#function to count the number of students who passed
def number_of_passed_stu(self):
   self.pcount = 0
   for i in self.result:
      if i == 'Pass':
         self.pcount = self.pcount + 1
      else:
         self.pcount
   return self.pcount

#function to count the number of students who failed
def number_of_failed_stu(self):
   self.fcount = 0
   for i in self.result:
```

```
        if i == 'Fail':
            self.fcount = self.fcount + 1
        else:
            self.fcount
    return self.fcount


#creation of object to access the class methods
obj_grade = Grade ()
obj_grade.get_marks()
obj_grade.calculate_mark()
obj_grade.total_marks()
obj_grade.average_marks()
print('Passed student count: ', obj_grade.number_of_passed_stu)
print('Failed student count: ', obj_grade.number_of_failed_stu)
```