CS 5007 Individual Project Assignment 3 (100 Points)
Due: 11:59 p.m. on 10/29/2021

Project Objective: The goal of this project is to develop and implement an application using the object-oriented programming paradigm.

Project Deliverables:

1. Submit a zip file that include your UML class diagram (**.pdf**), the hierarchy chart for the driver program (**.pdf**), and the five **.py** files to Canvas.
2. Provide the clear **comment documentations** to your code.

Note:

(1) This project is to be done by each student individually. No help besides the textbook, materials, and the instructor should be taken. Copying any answers or part of answers from other sources (including your classmates) will earn you a grade of zero.
(2) All programming conventions mentioned in class should be followed.
(3) You should test your program before submitting.
(4) Your program must be developed and implemented in the PyCharm IDE, or 10% of the graded score is deducted.
(5) Assignments are accepted in their assigned Canvas drop box without penalty if they are received by 11:59PM EST on the due date, or 10% of the graded score is deducted for the late submission per day. Work submitted after one week of its original due date will not be accepted.

Task:

Generally speaking, there are three types of exams, Multiple Choice exams, Technical Writing exams, and Analytical Programming exams, of courses offered by the IT department. The grade of each type of the exams determined is based on the following:

Exam Score	**Grade**
>= 90	A
>= 80	B
>= 70	C
>= 60	D
< 60	F

For a Multiple Choice exam, each correct question is worth 2 points. The exam score = [(CorrectNumOfMCQuestion * 2) / (TotalNumOfMCQuestion * 2)] * 100. The maximum score is 100.

For a Technical Writing exam, it contains three portions, Grammar, Sentence Structure, and Content. Each portion is worth 100 points, and the final score of the entire exam determined is based on the following percent distributions:

Portion	**Percent**
Grammar	30%
Sentence Structure	30%
Content	40%

For an Analytical Programming exam, it contains two sections, Short Answer and Programming. Each section is also worth 100 points, and the final score of the entire exam determined is based on the following percent distributions:

Section	**Percent**
Short Answer	30%
Programming	70%

Every class that you design must have a toString() method to return the current data of **each specific exam object** and a getExamGrade( ) method to return the final grade of an exam in each class.

Validations for the user inputs are required in this assignment.

For the driver program, **ITExamGrade**, it has two sub-functions under the main() function. One sub-function is used to accept **ONE exam object** and to fill the content of that object. Another sub-function is to display all the exam object information that is shown in the last screenshots.

All the output numerical values are in one decimal place.

All the functions of each class and the driver program are defined in the below tables. No addition functions are added.

ITExam Super Class Method List

The class has a constant, MAX_SCORE = 100.

Method	**Description**
__init__	Create the object and assign the initial value "None" to the exam title and 0 to the exam score.
setExamTitle	Check and assign the valid exam title to the instance variable and return True or else return False
getExamTitle	Return the exam title
setExamScore	Check and assign the valid score to the instance variable and return True or else return False
getExamScore	Return the exam score
getExamGrade	Return the grade based on the exam score
toString	Return the exam title and the exam score

MultipleChoiceExam Sub Class Method List

The class has a constant, POINTS_PER_QUESTION = 2.

Method	Description
__init__	Create the object and assign the initial value 0 to all the instance variables.
setTotalNumOfMCQuestion	Check and assign the valid number to the instance variable and return True or else return False
getTotalNumOfMCQuestion	Return the number
setCorrectNumOfMCQuestion	Check and assign the valid number to the instance variable and return True or else return False
getCorrectNumOfMCQuestion	Return the number
getExamGrade	Return the grade based on the exam score
toString	Return the exam title and the exam score

TechnicalWritingExam Sub Class Method List

The class has the following constants:

MAX_SCORE = 100
GRAMMER_PERCENT = 0.3
SENTENCE_PERCENT = 0.3
CONTENT_PERCENT = 0.4

Method	Description
__init__	Create the object and assign the initial value 0 to all the instance variables.
setGrammerScore	Check and assign the valid score to the instance variable and return True or else return False
getGrammerScore	Return the score
setSentenceStructureScore	Check and assign the valid score to the instance variable and return True or else return False
getSentenceStructureScore	Return the score
setContentScore	Check and assign the valid score to the instance variable and return True or else return False
getContentScore	Return the score
getExamGrade	Return the grade based on the exam score
toString	Return the exam title and the exam score

AnalyticalProgrammingExam Sub Class Method List

The class has the following constants:

MAX_SCORE = 100
SHORT_ANSWER_PERCENT = 0.3
PROGRAMMING_PERCENT = 0.7

Method	**Description**
__init__	Create the object and assign the initial value 0 to all the instance variables.
setShortAnswerScore	Check and assign the valid score to the instance variable and return True or else return False
getShortAnswerScore	Return the score
setProgrammingScore	Check and assign the valid score to the instance variable and return True or else return False
getProgrammingScore	Return the score
getExamGrade	Return the grade based on the exam score
toString	Return the exam title and the exam score

ITExamGrade

Function	**Description**
main()	(1) Display the exam menu to ask for the valid input choice. (2) If the input choice is valid, pass **the exam object** to the fillITExam(oneExamObject) function to create that exam object. (3) Ask for more IT exams. (4) If no more IT exams, call the displayResults(**allExamsList**) function to display all the exam contents that you ask for previously.
fillITExam(oneExamObject)	Ask for the users to enter the valid score for this exam object that passes into this function.
displayResults(allExamsList)	Display all the exam contents when the main() function calls this function.

** The program keeps running until there is no more IT exams entered, then the program displays all the exam contents that the user enters before. All the user's inputs are in the red color. See the below sample as the example on the next page.

Sample Output

```
IT Exam Type Menu:
1) Multiple Choice
2) Technical Writing
3) Analytical Programming
Enter your choice: 4

Sorry! No such choice. Please enter it again.

IT Exam Type Menu:
1) Multiple Choice
2) Technical Writing
3) Analytical Programming
Enter your choice: 1

Enter the Exam Title: Press the Enter Key Only
Sorry! The exam title is not valid. Please enter it again.

Enter the Exam Title: Introduction to Computing
Total Number of Multiple Choice's Questions: -50

Sorry! The number is not valid. Please enter it again.
Total Number of Multiple Choice's Questions: 50
Total Number of Correct Multiple Choice's Questions: 70

Sorry! The number is not valid. Please enter it again.
Total Number of Correct Multiple Choice's Questions: 45

More IT Exams (Yes/No)? Yes

IT Exam Type Menu:
1) Multiple Choice
2) Technical Writing
3) Analytical Programming
Enter your choice: 2

Enter the Exam Title: IT Applications in Business World
Score of Grammer Portion: -70

Sorry! The score is not valid. Please enter it again.
Score of Grammer Portion: 70
Score of Sentence Structure Portion: 185

Sorry! The score is not valid. Please enter it again.
Score of Sentence Structure Portion: 85
Score of Content Portion: 80

More IT Exams (Yes/No)? Yes
```

```
IT Exam Type Menu:
1) Multiple Choice
2) Technical Writing
3) Analytical Programming
Enter your choice: 3

Enter the Exam Title: Python Programming
Score of Short Answer Section: 165

Sorry! The score is not valid. Please enter it again.
Score of Short Answer Section: 65
Score of Programming Section: -85

Sorry! The score is not valid. Please enter it again.
Score of Programming Section: 85

More IT Exams (Yes/No)? No

Exam Title: Introduction to Computing
Exam Score: 90.0
Total Number of MC Questions: 50
Total Number of Correct MC Questions: 45
Final Grade: A

Exam Title: IT Applications in Business World
Exam Score: 78.5
Score of Grammer Portion: 70.0
Score of Sentence Structure Portion: 85.0
Score of Content Portion: 80.0
Final Grade: C

Exam Title: Python Programming
Exam Score: 79.0
Score of Short Answer Section: 65.0
Score of Programming Section: 85.0
Final Grade: C

Total Score of All Exams: 247.5
Average Score of All Exams: 82.5
```

Grading Criteria:

Checkpoint:	**Possible Points**
UML class diagram (.pdf)	5
Hierarchy Chart (.pdf)	5
Proper Naming Conventions	10
Program Documentation	10
ITExam Class	10
MultipleChoiceExam Class	10
TechnicalWritingExam Class	10
AnalyticalProgrammingExam Class	10
ITExamGrade Driver Program	15
Correct Above Sample Program Outputs	15
Total	100