

Lab 09
OOP – BSDS

Task 01: Create class Marks with following instance members:

- roll no (string)
- quiz 1 marks (float)
- quiz 2 marks (float)
- quiz 3 marks (float)
- presentation marks (int)
- assignment 1 marks (int)
- assignment 2 marks (int)

and following class members with zero initial value:

- max_quiz 1 marks (float)
- max_quiz 2 marks (float)
- max_quiz 3 marks (float)
- max presentation marks(int)
- max assignment 1 marks (int)
- max assignment 2 marks (int)
- sum of quiz 1 marks (float)
- sum of quiz 2 marks (float)
- sum of quiz 3 marks (float)
- sum of presentation marks(int)
- sum of assignment 1 marks (int)
- sum of assignment 2 marks (int)
- count (to count number of instances)

Write following instance methods/ functions:

- initialization function with 7 data members, assign values to instance members. Add 1 to count. Compare class level members with corresponding instance members and update max, if instance member is larger than class level member. Similarly, add every instance member (except roll no) into corresponding sum variable to get sum of all types of marks
- getter methods for all instance members
- str function to return string having all instance members in single line separated by tab character, some attributes require two tab characters, see the sample output at the end of this code

Write following class level member/ function to show statistics:

- show average marks for each category. Divide sum of each category with count
- show max marks in each category, max is already calculated by initialization functions

Run main function (written in “task1.py”) and show following output:

Roll no	Quiz1	Quiz2	Quiz3	Presentation	Assignment1	Assignment2
R1	9.5	8.5	6.75	90	45	70
R2	8.5	9.5	9.0	80	40	75
R3	7.0	9.0	8.75	85	42	80
Marks Statistics	Quiz1	Quiz2	Quiz3	Presentation	Assignment1	Assignment2
Average Marks	8.33	9.0	8.17	85.0	42.0	75.0
Max Marks	9.5	9.5	9.0	90	45	70

Task 02: You are provided with a text/ CSV (Comma separated values) file “result.csv”. This file has marks of 70 students in the way:

```
BCSF00M502,0,7.5,3,0,65,0
BCSF00M506,3,7.5,8,60,70,0
BCSF00M507,5,11,3,60,70,40
```

You can read this file line by line. Use split function with comma to get 7 separate values. Next you can pass to initialization function to create objects of class Marks. However, to pass values you must convert them into float and int according to their type given in task. Create following functions related to file handling:

- **read_data(empty list)**

Open file "result.csv" for reading. Read file line by line. On each line apply the following functions:

- use `rstrip` function to remove `\n` character from end of file
- use `split(',')` function to split attributes
- line will split into 7 attributes, pass them to create Marks object
- add object to list received as parameters

Finally, close the file

- **write_data(marks list)**

Open file "result.bin" for writing. Write all objects from list into binary file. Pack quiz 1 to quiz 3 using 'f'. Pack presentation marks and assignment 2 & 3 marks using 'h'. Concatenate roll no (already string) with quiz and other attributes and write complete string into file.

After calling this function successfully, check your file size, it should be 1960 bytes (70 * 28, 70 objects, each object size is 28. 10 bytes for roll no, 12 bytes for quiz 1 to quiz 3. 6 bytes for presentation, assignment 1 and assignment 2)

- **read_data_binary()**

Open file "result.bin" for reading. Read complete file into string. Next run a loop for step 28 (28 is size of one record). Create Marks object by splitting 28 bytes into roll no (first 10 bytes), quiz 1 through quiz 3 each 4 bytes using `unpack` with 'f'. Unpack presentation, assignment 1 & assignment 2 marks using 'h' each of 2 bytes. You will write something like:

`s[0:10]` for roll no

`s[10:14]` for quiz 1

`s[14:18]` for quiz 2

...

Create object and print to see data is successfully read from binary file.