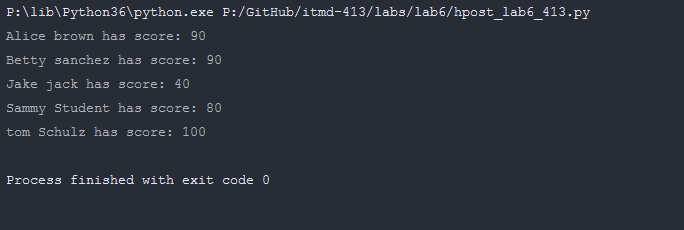
Henry Post

ITMD413

Lab 5, Payroll Processing



1. **import** time
3. \_\_info\_\_ = \
4. f"""
5. Henry Post,
6. Lab 06: OOP With Classes
8. ITMD413,
9. IIT Spring 2018,
10. Ran on {time.strftime("%c")}
11. """

14. **class** Student:
15. Scores = {}
17. **def** \_\_init\_\_(self, name, grade):  # Initializing constructor method
18. self.name = name
19. self.grade = grade
21. **def** \_\_str\_\_(self):
22. **return** f"{self.name}: {self.grade}"
24. **def** getScores(self):
25. answer\_key = []
26. # read into answer\_key list, the answer key from file
27. answer\_key = [line.strip() **for** line **in** open("answers.txt", 'r')]
29. student\_answers = []
30. # read into student\_answers list, student answers from file
31. student\_answers = [line.strip().split(',') **for** line **in** open("data.txt", 'r')]
33. total\_score = 100
35. '''''
36. Finish the processing logic below marked within the commented lines to
37. correctly grade each student by creating a loop(s) to correctly score
38. students using the lists created above (namely the answer\_key list and
39. the student\_answers list) by matching each of the student answers to
40. each answer key item
42. Each incorrect answer deducts 10 points from the 'total\_score'
43. variable shown (intialized) just above
44. '''
46. # ---start your loop processing logic here---#
48. student\_answers\_d = {}
50. **for** answer **in** student\_answers:
51. name = answer[0]
52. answers = answer[1:]
53. student\_answers\_d[name] = answers
55. **for** i **in** range(len(student\_answers\_d[self.name])):
56. **if** student\_answers\_d[self.name][i] != answer\_key[i]:
57. total\_score -= 10
59. # ---end your loop processing logic here---#
61. Student.Scores[self.getName()] = total\_score;
63. **def** getName(self):
64. **return** self.name;
66. @staticmethod
67. **def** sortDict():
68. **return** sorted(Student.Scores.items());

71. student\_objs = [
72. Student(line.split(',')[0], -1) **for** line **in** open("data.txt", 'r')
73. ]
75. **for** index **in** range(len(student\_objs)):
76. student\_objs[index].getScores();
78. sortList = Student.sortDict();
80. **for** k, v **in** sortList:
81. **print**(k, "has score:", v)