

## Assignment 02

### CSIS 2175 – Section 004

Each question in the assignment carries 10 marks. The assignment contributes 5% to the overall grades. Students are required to submit the assignment to blackboard not later than **Feb 18, 2020 15:30 PST**. **NO LATE SUBMISSION** will be accepted. You may submit your work multiple times, but only the last submission will be graded.

Assignment could undergo a similarity check. If plagiarism is detected or students are found to have copied from each other, marks would be deducted.

#### **Grading:**

- Correctness of the program: 80%
- Programming style/comments/clarity: 10%
- Correctness of output format: 10%

#### **Problem 1:**

Create a TaxPay class with fields that hold a taxpayer's Social Security number, last name, first name, street address, city, state, zip code, annual income, tax. Include a constructor that requires arguments that provide values for all the fields except tax. The constructor calculates the tax liability based on annual income and the percentages in the following table. Include a display method that displays all the information data. Save the file as **TaxPay.java**.

Annual Income (\$)	Tax Rate (%)
0-41725	5.06
41725.01-83451	7.70
83451.01-95812	10.50
95812.01-116344	12.29
116344.01-157748	14.70
157748.01-220000	16.80
220000-More	20.50

Create an application that prompts a user for the data needed to create a TaxPay for a resident. Continue to prompt the user for data as long as any of the following are true:

- The Social Security number is not in the correct format, with digits and dashes in the appropriate positions; for example, 999-99-9999.
- The zip code is not six digits with format A9A9B9.
- The annual income is negative.

After all the input data is correct, create a TaxPay object and then display its values. Save the file as **TaxMain.java**.

#### **Sample Output**

```
Console [X]
TaxMain [Java Application] C:\Program Files\
Enter Social Security number
111-11-1111
Enter Annual Income
34000
Enter Zip Code
040555
Enter Social Security number
```

```
Console [X]
TaxMain [Java Application] C:\Program Files\Java\
Enter Social Security number
123-11111111
Enter Social Security number
```

```
Console [X]
<terminated> TaxMain [Java Application] C:\Program Files\Java\j
Enter Social Security number
232-88-9999
Enter Annual Income
850000
Enter Zip Code
D3E5R5
Enter Last name
John
Enter First name
Mary
Enter Street name
Robert
Enter city name
Abbotsford
Enter state name
BC

Details of the Tax payer
=====
Social Security Number: 232-88-9999
Last name: John
First Name: Mary
Street: Robert
City: BC
State: D3E5R5
Annual Income: 850000.0
Tax: 17425.0
```

```
Console
TaxMain [Java Application] C:\Program Files\Java
Enter Social Security number
111-11-1111
Enter Annual Income
-8900
Enter Social Security number
```

## Problem 02:

Create a class named Student that contains fields to hold methods for setting and getting a Student Id, Name, and marks. Save the file as **Student.java**.

Write an application that asks for the number of students to be recorded and instantiates Student objects. It should prompt the user for values for the Student Id, Name and Marks. Then prompt the user to enter the field the Student records should be sorted by—Student Id, Name, or Marks. Perform the requested sort procedure and display the Student records in descending order. Save the file as **StudentSort.java**.

The application should continue to ask for more sorting until user enter 1 to continue. To exit enter any integer other than 1.

Number of student records to be entered

5

Enter student data

Enter data for Student 1

Enter student Id

401

Enter student marks

44.5

Enter student name

John

Enter data for Student 2

Enter student Id

204

Enter student marks

33.5

Enter student name

Mary

Enter data for Student 3

Enter student Id

305

Enter student marks

88.6

Enter student name

Kim

Enter data for Student 4

Enter student Id

102

Enter student marks

45.6

Enter student name

Fred

Enter data for Student 5

Enter student Id

503

Enter student marks

66.8

Enter student name

Tom

Information about the student records entered:

Student Id: 401 Student Name: John Student marks: 44.5

Student Id: 204 Student Name: Mary Student marks: 33.5

Student Id: 305 Student Name: Kim Student marks: 88.6

Student Id: 102 Student Name: Fred Student marks: 45.6

Student Id: 503 Student Name: Tom Student marks: 66.8

Enter 1 to sort records by Student Id, 2 to sort by Name, 3 to sort by marks

1

Information about the student records sorted by Student ID

Student Id: 503 Student Name: Tom Student marks: 66.8

Student Id: 401 Student Name: John Student marks: 44.5

Student Id: 305 Student Name: Kim Student marks: 88.6

Student Id: 204 Student Name: Mary Student marks: 33.5

Student Id: 102 Student Name: Fred Student marks: 45.6

Do you want to display sorted student records again? Press 1 to continue

1

Enter 1 to sort records by Student Id, 2 to sort by Name, 3 to sort by marks

2

Information about the student records sorted by Student Name

Student Id: 503 Student Name: Tom Student marks: 66.8

Student Id: 204 Student Name: Mary Student marks: 33.5

Student Id: 305 Student Name: Kim Student marks: 88.6

Student Id: 401 Student Name: John Student marks: 44.5

Student Id: 102 Student Name: Fred Student marks: 45.6

Do you want to display sorted student records again? Press 1 to continue

1

Enter 1 to sort records by Student Id, 2 to sort by Name, 3 to sort by marks

3

Information about the student records sorted by Student Marks

Student Id: 305 Student Name: Kim Student marks: 88.6

Student Id: 503 Student Name: Tom Student marks: 66.8

Student Id: 102 Student Name: Fred Student marks: 45.6

Student Id: 401 Student Name: John Student marks: 44.5

Student Id: 204 Student Name: Mary Student marks: 33.5

Do you want to display sorted student records again? Press 1 to continue

0