# SEA400- Assignment 1

| Total Mark: | 10 marks |
| --- | --- |
| Submission file(s):  (do **NOT** zip files) | * Team\_Contract.docx * Asg1.docx * student.py * bestSemester.py   Please do NOT zip the files for submission. |

Please work in **groups** to complete this lab. This lab is worth 3% of the total course grade and will be evaluated through your written submission, as well as the lab demo. During the lab demo, group members are *randomly* selected to explain the submitted solution. Group members absent during the lab demo will lose the demo mark.

Please submit the submission file(s) through Blackboard. Only one person must submit for the group and only the last submission will be marked.

## **Part I: The Team Contract**

1. In this course, you will be doing lots of group work. To make sure this will be a good experience for all, let’s work on preparing a team contract. This contract must contain “Team Procedures”, “Team Expectations” and “Consequences”. See the following links for samples and ideas (samples only- your contract does not have to be exactly the same):

* Guidelines for writing team contracts- University of Arizona: <http://math.arizona.edu/~kerimar/Team%20Contract.doc>
* Team Contract- MIT:

[http://web.mit.edu/6.005/www/fa15/projects/abcplayer/team-contract/#team\_contract](http://web.mit.edu/6.005/www/fa15/projects/abcplayer/team-contract/" \l "team_contract)

Submit a signed copy of the contract as Team\_Contract.docx. Digital signatures are enough (add your names at the end of the document), as long as you all have reviewed the agreement and agree to it.

## **Part II: Software Setup**

1. Follow the instructions in **SEA400\_softwaresetup.pdf** to set up the required software for this course.

## **Part III: Review Python Basics**

1. Review Python Basics from this link:

<https://inst.eecs.berkeley.edu/~cs188/sp22/project0/#python-basics>

1. Download and unzip tutorial.zip: <https://inst.eecs.berkeley.edu/~cs188/sp22/assets/files/tutorial.zip>
2. In Anaconda Navigator, go to Environments tab and click on the triangle next to SEA400. Choose “Open Terminal”.

A screenshot of a computer

Description automatically generated

Change directory to the unzipped folder, for example (this may be different from your path):

cd C:\SEA400\tutorial

Keep this terminal open and run code here. For example, you can check if you correctly answered Question 1 by running this code:

python autograder.py -q q1

1. Follow instructions to solve and test questions 1 to 3 from above link (Project 0).
2. If you want to use the debugger in VS code, see:

[Debugging in Visual Studio Code](Debugging%20in%20Visual%20Studio%20Code) <https://code.visualstudio.com/Docs/editor/debugging>

For example, to run the debugger with *autograder.py* test samples for question 1, change *launch.json* to include

"args": ["-q", "q1"],

Paste screenshots of the results of the following commands here:

python autograder.py -q q1

Result:

A black screen with white text

Description automatically generated

python autograder.py -q q2

Result:

A screenshot of a computer

Description automatically generated

python autograder.py -q q3

Result:

A screenshot of a computer

Description automatically generated

python submission\_autograder.py

Result:

A screenshot of a computer

Description automatically generated

## **Part IV: More Python**

1. Inspired by *shop.py*, modify *student.py* (given in this workshop) to complete the following for class *Student*:
   * The initialization method, takes the *name* of the student, as well as *courseUnits*, a dictionary with keys as course strings and units for values e.g. {'math':4.0, 'science': 3.0, 'art':2.0} and sets the related class variables.
   * Method *getCourseUnits(self, course)* which looks up ‘course’ and returns units of 'course', assuming 'course' is in *courseUnits* dictionary or None otherwise
   * Method *getGPA(self, gradeList),* which given *gradeList*, a list of (course, grade) tuples, calculates and returns GPA, only including the grades of courses that this student has registered for (those available in *courseUnits*). If none of the courses are in *courseUnits*, it will return *None*.

For example, for above student, if *gradeList* is [('math',85.50), ('art',92.0)], it returns 87.67 (note the rounding), i.e. (85.50 \* 4.0 + 92.0 \* 2.0) / (4.0 + 2.0)

And if *gradeList* is [('science',80.00), ('geography',95.0]), it returns 80.00, since ‘geography’ is not in this student’s *courseList.*

1. Inspired by *shopSmart.py*, modify *bestSemester.py* (given in this workshop) to complete the following:

* Function *highestGPA(semGradeList, astudent)* which calculates the highest term (semester) GPA given a *semGradeList*, a list of semester grade lists and a student.
* Given the student and grade lists (don’t change these), complete the print statements in main, so running this code results in the following output:

|  |
| --- |
| Student Jim registered  GPA for semester 1 for Jim is 83.0  Student not registered for sea100  GPA for semester 2 for Jim is 88.09  Student not registered for sea100  The highest achieved term GPA by this student is 88.09 |

## **A screenshot of a computer Description automatically generated**

## **Part V: Group work**

1. Complete this declaration by adding your names:

We, Chet, Kasra, Nahaeli, Atem-Ako, declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. We have not copied any part of this assignment, manually or electronically, from any other source including web sites, unless specified as references. We have not distributed our work to other students.

1. Specify what each member has done towards the completion of this work:

|  |  |  |
| --- | --- | --- |
|  | Name | Task(s) |
| 1 | Sthapanavichet Long | All parts |
| 2 | Atem-Ako Eyong Atem | All parts |
| 3 | Nahaeli Brunder | All parts |
| 4 | Kasra Bina | All parts |