**Authors:** Hetvi Shastri, Ankur Aditya and Anirudh Sabnis

**Course number and name:** Introduction to Algorithms(COMPSCI 311)

**Content Area**: Data Structure and Algorithms

**Title of RLO:** Programming assignment – The Atomic Nature of Matter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objectives/Goals:**  Design programming assignments for the Introduction to Algorithms (CS 311) course. The programming assignments will test the student’s ability to use concepts learnt in the class to design and implement efficient algorithms for real-world problems. | | | | |
| **Knowledge:** *(Students should know...)*  Knowledge:   1. Concepts: The students will apply the concepts/techniques learnt in the class such as graph algorithms and recursion to solve the programming assignment. | | **Skills:** *(Students should be able to…)*  Skills:   1. Coding skills: the programming assignment is to be turned in in Java or Python. 2. Debugging: the programming assignment is complex enough for the students to spend considerable time debugging the code. 3. Code optimizations: the designed test cases will check the runtime of the code and will fail if the runtime exceeds a certain threshold. This will force students to write optimized code. | | |
| **Dispositions:** *(Students should adopt...)*   1. Algorithmic thinking: The students will need to apply/choose techniques learnt in the class to come up with efficient algorithms with the least possible complexity. 2. Writing optimized code: The test cases will fail if the runtime is greater than a certain threshold, thus, forcing the students to optimize their code. | | | | |
| **Vocabulary** | | | **Resources Needed** | |
| **Content Vocabulary** *(definition)*  Avogadro number, Brownian motion, Boltzmann constant, pixel, luminance | | | **Content Resources** *(books, articles, websites, handouts etc.)* CLRS: Introduction to Algorithms | **Web resources:**  <https://www.cs.princeton.edu/courses/archive/spring17/cos126/assignments/atomic.html> |
| **Computer Science Vocabulary** *(Vocabulary Definition)*  Python, Java, Recursion, Graph algorithms | | | **Hardware:**  Laptop | **Other Manipulatives:** |
| **Plan for Instruction** | | | | |
| **Crafting:** *(Teacher Lead Instruction) Presentation of new material – Include script for lesson.* | | | | |
| **TEACHER ROLE** | **STUDENT ROLE** | | | **Checks for Understanding** |
| **What is the teacher doing?**   1. The teacher will teach the students the required algorithms for the assignment. 2. The teacher will explain the assignment in the discussion class 3. The teacher will explain about the academic policy on plagiarism. 4. He will explain about the grading. | **What are the students doing?**   1. The students will attend the class. Try to understand the concepts. Review the concepts by going through the above mentioned content resources. 2. The students will form groups of 3-4 to discuss the possible solutions. | | |
| * 1. Teacher will try to ask some engaging questions in the class. He can maybe take some feedback using poll to check whether majority of students have understood it.   2. In the discussion class the teacher will interact all the students to make sure the students have understood the programming assignment.   3. The teacher can check for academic honesty teacher by looking at the similarity report on gradescope. |
| **Composing Meaning:** *(Independent) Students working by themselves, with partners, or in groups. –* | | | | **Checks for Understanding** |
| **What is the teacher doing?**   1. Teacher will answer questions/concerns/doubts on piazza and office hours. | **What are the students doing?**   1. Students can ask doubts without hesitation on piazza/ office hours. 2. Students will write code and submit code on gradescope and ensure it passes the test cases. 3. Students will also submit written subjective report describing their approach. 4. Student won’t interact, collaborate, ask for help from any other student due to the academic policy on plagiarism. | | | 1. There will be office hours to get an understanding about how efficiently students are currently working on the assignment. 2. There will be many test cases to cover from simple scenarios to complex scenarios. |
| **Processing:** *(How will students reflect on today’s lesson and make connections to the objectives/goals and essential questions?)* | | | | **Checks for Understanding** |
| **What is the teacher doing?**   1. The teacher will post the slides and recorded lectures taught in the class for reference. 2. The teacher will link the assignment material and the concepts taught the class. 3. The teacher will provide sample solution after submission for reference. 4. The teacher will grade the programming and provide feedback. | **What are the students doing?**   1. Students will review class slides and recorded lecture before/while attempting the programming assignment. 2. Students can raise regrade requests. 3. Students will review the solution to check their understanding. 4. Students can ask further questions for clarification after looking at the solution. | | | 1. If necessary, along with automatic grading the assignment can be graded manually. |
| **Assessment/grade:**  We will have two grading methods. One will be manual and other will be automatic (passing test cases on gradescope).  We will also give 50% weightage for the subjective report. | | | | |