Assignment Report for [Assignment 04]

Course and Section	[CSC].[215]
Assignment Name	[Assignment 04]
Due Date and Time	[March 15] @ [11:59 pm]
First Name and Last Name	[Arjun] [Parujanwala]
SFSU Email Account	[aparujanwala@gmail.com]
First Name and Last Name of Teammate	[Arjun] [Parujanwala]
SFSU Email Account of Teammate	[tmueller@sfsu.edu]

PART A

Question Description and Analysis:

This part of the assignment asks that I store the English alphabet in both a 2D array using shorthand notation and a 2D ragged/jagged array using the multi-step approach. These arrays must have 5 rows and at least 2 columns in each row then I must display them into an identical output given. Then in ⅓ a page or more, I need to explain in detail my understanding of the problem and another ⅓ of my analysis of my program.

Answer:

This is my answer…

1. The problem in the most simplest terms is asking me to store the alphabet in 2 different types of arrays and 2 different methods of achieving so. What I notice about the desired output is that both arrays need to be right aligned and also contain a service description of what the program is doing. Then, getting into more specifics, the client asks that there should be no empty stored characters in my array and creates a 1 method that displays the alphabet stored in both arrays. Judging from the desired output again, the desired output

also wants me to display them in a specific way where there are 7 letters in the first row, 4 in the second etc.

2. Yes, my program does indeed compile as expected and I have looked over the requirements and I have found that my code in fact uses a 2D array and the shorthand notation and also uses a 2D ragged array and the multi-step approach to store the 26-letter English alphabet. Both arrays have 5 rows and at least 2 columns in each row. It also has a method that displays both arrays and also matches the sample output. Not only that, the output has no empty characters. What worked is the print because arguably it is the hardest part of the program and I had to use 2 different 4 loops as well as an if statement to replace empty characters. What could've been done better is how I stored the ragged/jagged array because I feel like there was definitely a better way in storing them.

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…

PART B

Question Description and Analysis:

This part of the assignment asks that I ask the user for specific data types and then store it in a given 2D Array, “Object[][] inputDatabase = new Object[4][3];” and then store I must retrieved the data stored in that array and print out a Data Type View and a Data Value View. Then in ⅓ a page or more, I need to explain in detail my understanding of the problem and another ⅓ of my analysis of my program.

Answer:

This is my answer…

1. My understanding of the problem is that it basically asks me to asks the user for data of specific types and with those types I need to make a data type view which means if the user provides me with an integer then I have to print out “Integer” and then it also asks me to make a data value view in which I just print out the user input. My plan is to make four 1D arrays for the user inputs and then store the values in the array into the given object. Then, I will then use the object to go through a getClass method in order to print out the Data Type View and then use for loops to print out the Data Value View.
2. My program does indeed compile as expected and it does require the requirements such as styling, the usage of the given code, user inputs in the first 4 lines of the program. Not only that, it also automatically detects the type of stored data for display and also the data value view retrieves that data for display. I think everything works but what could be improved is again how I stored the data into the given code because it just seems like I hardcoded everything into the Object array and there is definitely a better way to do it.

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…

PART C

Question Description and Analysis:

This part of the assignment asks that…to create a Java program to track growth of a plant.

Answer:

This is my answer…

1. Problem Analysis and Problem-Solving: For this problem, it is asking for a program to take in a minimum and maximum temperature for a plant and a minimum rainfall value for the plant and outputting temperature, rainfall, plant growth, plant height values for all months of the year. Calculating those values with average temperatures and rainfall for each month. The client wants an easy to read table that displays all the values in a nice formatted manner with the highest plant's height highlighted on the table. As well as of course calculating the client's plant's growth and height for the 12 months from the average temperature and rainfall for each month. Some important details I noticed was that there was a plus sign for any positive integer on the plant growth column as was as equal spacing for all the columns. A space for every single integer on the index column, as well as a highlighted word "MAX" next to each of the greatest plant height value. And the program to ask for minimum and maximum temperature values and minimum rainfall values of the plant from the user. I solved these problems one by one by creating a method specifically for that one problem. Some problems were being able to display avg temp, rainfall values on the table. I made two methods that took in what month number it was based on the variable "i" and displayed the average temperature and rainfall for that specific month which worked well and displayed all 12 months average values perfectly. Methods were a great way to organize and design my way around the problems. The important elements are definitely the formatting of the output and of course the

calculations of plant growth and plant height being made and outputted correctly. I will pay attention to how the code is organized and if it needs comments for the person reading the code.

3. Result Analysis and Future Development: After creating and running the program, it compiles and outputs the correct values perfectly. I checked all the outputs if they were formatted and lined up correctly which they did. All the values are displayed correctly including the calculations of plant growth and height. It works because the methods created for plant growth and height go through each month of the average temperature and rainfall and calculates the growth and height for that month and returns and prints it onto the table. The height can take the growth values and add or subtract it to the overall plant height which is why all the values add up perfectly and are calculated correctly on each month. So far everything works and there are no problems in which something does not work. The only thing to be improved is organizing and adding comments to the code for someone to be able to read it more easily. And if the client is not happy with the program, we can format the data to be displayed in a more user friendly manner. For my 5 meaningful methods, I created the following methods: getgrowth, getheight, getavgTemp, getavgRain, and getNumberWithPlusSign. These methods were crucial in building the program. For getgrowth, it took in mintemp, maxtemp, minrain, avgtemp, and avgrain. Used all those values to calculate how much the plant would grow and return that value which is very meaningful to the program since it gives us the data needed to output the table. Getheight took growth and height and returned the value of the height of the plant displaying how much the plant has grown or shrunk to show the user what will happen in the coming months. getavgTemp took in variable “i” and would return a value from the avgTempData array. Supposed we want the average temperature for March, “i” would be 3 because March is the 3rd month and the

method would take in that value and return the average temperature for that month which is crucial in calculating the growth of the plant as well as display those average temperatures on the table. GetavgRain method works similarly to getavgTemp method in which it also takes value i and return the value of the average rain will be for that month which is a value also crucial for calculating the plant's height and growth. getNumberwithPlusSign method takes the plant growth values and add a plus sign to any positive value which is important to the user to identify easily if the plant is growing or shrinking for that month and adds an overall positive experience to anyone using this program.

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…

PART D

Question Description and Analysis:

This part of the assignment asks that…to create a student program.

Answer:

This is my answer…

1. Problem Analysis and Problem-Solving: The problem is asking us to be able to enter student information from the user which can be used to create multiple different student's information and display them correctly. But not only that, we must be able to update the student's information if the user wishes to. After the update, it must show the updated students' information. The requirements of the program is that we must use the given constructors and methods to be able to update and set the student's information given by

the user. But also the program must be able to detect which student the user wants to update his/her information. The important details from the desired outputs I see, is that the program must be able to detect which student is being updated regardless for capitalization. As well as correct spacing and formatting. After updating, the output must show the updated information. I plan to solve this problem by first writing the code necessary in getting the input from the user and then going one by one for each problem and solving them in order to get the right updated information and outputs from the user. I will organize these by sections in taking in the user input, setting the name and gpa of the students, and displaying them, as well as an updating input section and an updated output section. That way the code is easy to read which is something I will pay attention to which is crucial for interviews and problem solving.

3. The problem compiles perfectly and outputs the correct information of each student as well as the updated student information. It satisfies all the client's requirements and is able to update any specific student the client wishes and shows the updated student information. The program works in setting the name and gpa of every new student put into the problem. It gives them unique names and values for each student if needed to be updated, the program allows the user to do so. It works because the code is written so that if a student needs updated information, it will take the updated information from the user and set it for that student giving it new updated values in which it will be displayed for the user. Everything works perfectly and do not see any problems. Some things that can be improved is adding comments to the code for anyone reading the code to easily identify which code does what. I plan to improve the program by displaying the student's information as well as the updated information in a more user friendly fashion. For my 5 meaningful methods, I used: getname,setname, getgpa, setgpa, and toString. Both getname and

getgpa were crucial in making the program as they can get and return the values of any specific student that was put into by the user. Which is needed for displaying each of the student's information in the output of the program. We can easily show it and reuse the code by using the getgpa and getname methods. Setgpa and setname methods were needed because they were able to grab the input of the user and set the values of the new student made in the program. Thus we could have multiple students made and each would have their own set name and gpa which could be displayed by the two previous explained methods. These two methods were also useful in updating the student's information. toString method was very useful as it could get the specific name and gpa of each student as well as the updated information and display it in the specified format correctly each time. This made it very easy as I could reuse this method and display all the different and updated student's information as many times as I wanted.

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…

PART E

Question Description and Analysis:

This part of the assignment asks that I create a 3D String[4][3][2] array containing 24 items and one 1D[Semester] array containing 4 items.

Answer:

This is my answer…

Container 3D:

1. How I loaded and organized the information from the 3D array:

   I loaded and organized the information by first making a loadSemesterData method which contains parameters that contain all the 1D arrays so I can access them inside the method. Then, I of course made the 3D array and set the desired boundaries. Then individually row by row, I would set the value ex:(semester[0][0][0] = semester01[0];) until I finish filling them into their corresponding semester. Then it would pretty much repeat the same process but the only difference would be the first bracket when I'm changing the row and also the semester that I am trying to retrieve the value from. Then, I would simply just return semesters so that it contains the data information in a structured format.

2. How I printed it was basically I loop through the semester, then print the semester information, then still in the same loop I loop the course, then loop through the information within each course, then print course information then I format and also just print a new line for the next semester.

Container OOP:

1. I loaded and organized the semesters by simply creating the 1D Semester[] array with a boundary of 4 and then manually loaded the array by doing EX: semesters[0] = new Semester(semester01); and did that for the rest of the semesters.
2. I printed by simply turning the arrays into strings and using a for loop that takes the length of the array and print each semester 1 by 1.

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…