Module 10 In-Class Practice 4

DMA of an Array of Nested Structures

### Student Time Management



Write a program (name the source file **mod10d.cpp**) that will calculate for five students if they should take a part-time job or not based on how many credit hours they are currently enrolled in.

### STRUCTURES

Create two structures named Course and Student. The Course structure should be nested in the Student structure.

**Course:**

* Number of courses (integer)
* Pointer to an integer (this will eventually be an integer array that is dynamically allocated)

**Student:**

* Student name (string)
* House (string) – this is the name of the student’s computer science house – Borg, Dijkstra, Hopper, Lovelace, Neumann, or Turing
* Course info (this is a variable member that is of data type Course.

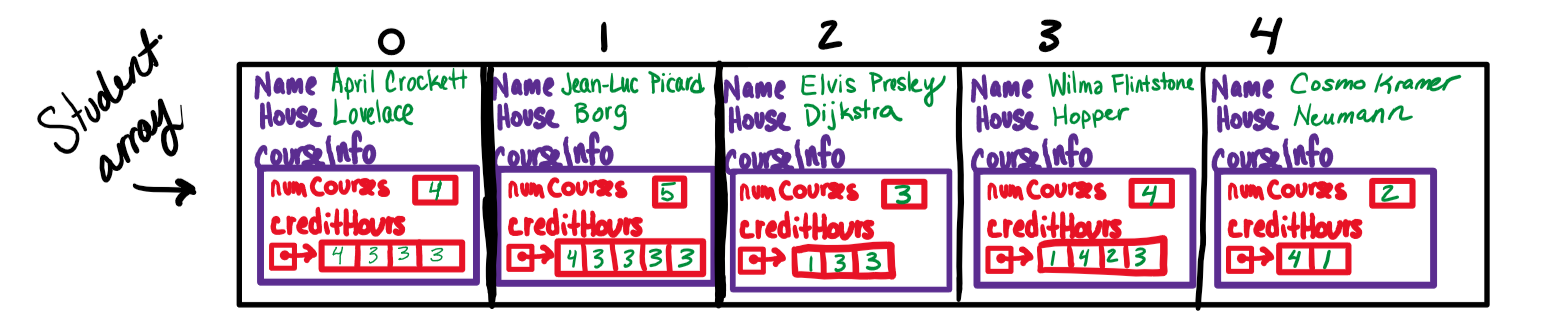
### Main Function

1. Create a 5-element **Student** array.
2. Print out “Which of the five students can work in addition to going to school?”.
3. Call the **enterData** function, sending the **Student** array.
4. Loop through each of the five students and in the loop, call the **calculateHours** function, sending the particular **Student** element. If the total (integer) returned from **calculateHours** is larger than 30, print “[name] is already going to class and studying [total hours returned from function] hours, so they should not take a job this semester.” If the total returned from **calculateHours** is 30 hours or below, then print “[name] is going to class and studying [total] hours, so they could take a part-time job.”
5. Last, release any dynamically allocated arrays.

### enterData Function

1. Print out “Enter each student’s information”.
2. Loop through each student in the Student array and get the student’s name, house, number of classes.
3. Once you get the number of classes, dynamically allocate an array of the size of the number of classes and the pointer created in the Course structure should point to this array.
4. Loop through each Course array element and enter in the credit hours for each course.
5. Refer to sample output for the format of what should be printed to the screen during this function.

Once the data is stored, it will kind-of look like this in memory:



### calculateHours Function

The **calculateHours** function accepts just a single **Student**, calculates the total number of study + class-time hours for all classes, and returns this total.

Create an integer variable to hold the total hours.

In a loop, go through the credit hours array and for each course, you will calculate the study time as 3 hours for each hour of class time, plus add in the class time. So, for example, if a student had 4 courses and the credit hours stored in the credit hours array is 4, 3, 3, and 3, the total would end up being 52 hours.

Course 1 = 4 hours class + 4\*3 hours of study time = 16 hours

Course 2 = 3 hours class + 3\*3 hours of study time = 12 hours

Course 3 = 3 hours class + 3\*3 hours of study time = 12 hours

Course 4 = 3 hours class + 3\*3 hours of study time = 12 hours

Total for all courses = 16 + 12 + 12 + 12 = 52 hours

#### sample output

User input is highlighted in **yellow**.

**Which of the five students can work in addition to going to school?**

**Enter each student's information.**

**\*\*\*\*\*\*\*\*\*\*STUDENT 1\*\*\*\*\*\*\*\*\*\***

**Name: April Crockett**

**House: Lovelace**

**Number of classes this semester: 4**

**Credit hours for each course:**

**Course 1 - 4**

**Course 2 - 3**

**Course 3 - 3**

**Course 4 – 3**

**\*\*\*\*\*\*\*\*\*\*STUDENT 2\*\*\*\*\*\*\*\*\*\***

**Name: Jean-Luc Picard**

**House: Borg**

**Number of classes this semester: 5**

**Credit hours for each course:**

**Course 1 - 4**

**Course 2 - 3**

**Course 3 - 3**

**Course 4 - 3**

**Course 5 - 3**

**\*\*\*\*\*\*\*\*\*\*STUDENT 3\*\*\*\*\*\*\*\*\*\***

**Name: Elvis Presley**

**House: Dijkstra**

**Number of classes this semester: 3**

**Credit hours for each course:**

**Course 1 - 1**

**Course 2 - 3**

**Course 3 - 3**

**\*\*\*\*\*\*\*\*\*\*STUDENT 4\*\*\*\*\*\*\*\*\*\***

**Name: Wilma Flintstone**

**House: Hopper**

**Number of classes this semester: 4**

**Credit hours for each course:**

**Course 1 - 1**

**Course 2 - 4**

**Course 3 - 2**

**Course 4 - 3**

**\*\*\*\*\*\*\*\*\*\*STUDENT 5\*\*\*\*\*\*\*\*\*\***

**Name: Cosmo Kramer**

**House: Neumann**

**Number of classes this semester: 2**

**Credit hours for each course:**

**Course 1 - 4**

**Course 2 – 1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**April Crockett is already going to class and studying 52 hours, so they should not take a job this semester.**

**Jean-Luc Picard is already going to class and studying 64 hours, so they should not take a job this semester.**

**Elvis Presley is going to class and studying 28 hours, so they could take a part-time job.**

**Wilma Flintstone is already going to class and studying 40 hours, so they should not take a job this semester.**

**Cosmo Kramer is going to class and studying 20 hours, so they could take a part-time job.**