ASSIGNMENT 2 CPP

Prepared by Dr Han Duy Phan.

Submission instruction: Create a folder (A2_StudentName_StudentID) that contains 6 sub folders for each part. Each sub folder contains all the solutions for the corresponding tasks. Compress the A2_StudentName_StudentID folder and submit A2_StudentName_StudentID.zip to Moodle.

PART 1

Implement a simple C++ student management program. For this part, you only need to keep the information of each student's name.

(1) Use C++ array data structure to store the list of all students' names.

(2) Implement a function that allows users to add a new student to the list of all students.

(3) Implement a function to display the list of all students.

(4) Implement a function to remove a student from the list of all students based on the name input from the user. If the student name does not exist in the current list of all students, inform the user about that.

(5) Use C++ WHILE loop to create a text-based menu that allows users to choose the above functionalities or to exit the program.

PART 2

Improve the simple C++ student management program from Part 1. For this part, you need to keep the information of each student's name and their score in two separate dynamically allocated arrays using pointers.

(1) Use C++ dynamic array data structure using pointers to store two lists of all students' names and students' scores, 1 list for students' names & 1 list for student' scores.

(2) Implement a function that allows users to add a new student and the corresponding score to the two above lists.

(3) Implement a function to display the list of all students and the scores.

(4) Implement a function to display the best student(s)'s name(s) and their score(s) (there could be more than one student who has the same highest score).

(5) Use C++ WHILE loop to create a text-based menu that allows users to choose the above functionalities or to exit the program.

PART 3

Improve the simple C++ student management program from Part 2. For this part, you need to keep the information of each student's name and score in a class and maintain the list of student objects in a dynamically allocated array using a pointer.

(1) Create the Student Class that stores name and score. Make sure to have a constructor and destructor.

(2) Use C++ dynamic array data structure using pointers to store a list of all the student objects.

(3) Implement a function that allows users to create and add a new student object to the list of students (remember to use constructor through 'new' keyword).

(4) Implement a function to display the list of all students and their scores.

(5) Implement a function to display the best student(s)'s name(s) and their score(s) (there could be more than one student who has the same highest score).

(6) Implement a function to remove a student from the list based on the name input (remember to use destructor through 'delete' keyword)

(7) Use C++ WHILE loop to create a text-based menu that allows users to choose the above functionalities or to exit the program.

PART 4

Improve the simple C++ student management program from Part 3. For this part, you need to create a new class University. The two main private member attributes of this class are the university name and the list of the students. Bring all the features of Lab 3 to the University class as methods; also keep the WHILE loop in the main function. Make sure to have appropriate scope (private, protected, public) for the methods of the

University class; some of them need to be public so they can be invoked from the WHILE loop in the main function.

PART 5

Improve the simple C++ student management program from Part 4. For this part, you need to create two new classes UniStudent and CollegeStudent derived from the existing class Student. While both types of students have similar attributes like names, day of births, school names, course names; there are differences between uni students and college students. Uni students have 8 semesters, and each semester has 4 courses, and each course has 3 assignments, 2 tests and 1 exam. College students have only 4 semesters, each semester has 3 courses, and each course has 1 assignment, 1 test and 1 exam.

You need to appropriately decide the common attributes/methods and the distinguished attributes/methods for the parent class Student and its two child classes UniStudent and CollegeStudent.

PART 6

Improve the simple C++ student management program from Part 5.

From Part 5, you have already created two new classes UniStudent and CollegeStudent derived from the existing class Student.

For this Part 6, you need to utilize C++ polymorphism to implement the three functions DoAssignment, TakeTest and TakeExam for all the students. Essentially, calling these functions will randomly assign scores for all the assignments, tests and exams of all the courses of all the assignments of a student. However, the implementation for these functions are different for UniStudent and CollegeStudent due to their different numbers of semesters, courses, and assessments. Since this program manages all the students in one dynamic array of pointers to the base class Student, polymorphism is required to have correct function calls for different types of students.