

## 2003 AP<sup>®</sup> COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

1. Information about a `College` object includes its name, its tuition, and the region in which it is located.

The following class declaration will be used to store information about a college.

```
class College
{
    public:
        apstring Name() const;
        // returns college name

        apstring Region() const;
        // returns region of college

        int Tuition() const;
        // returns college tuition

        void SetTuition(int newTuition);
        // set college's tuition to newTuition

    // ... constructors, other member functions and data not shown
};
```

The following class declaration will be used to store information about a group of colleges.

```
class CollegeGroup
{
    public:

        void UpdateTuition(const apstring & collegeName,
                           int newTuition);
        // precondition: collegeName exists in this CollegeGroup
        // postcondition: tuition for collegeName is changed to newTuition

        apvector<College> GetCollegeList(const apstring & region,
                                         int low, int high) const;
        // precondition: low <= high
        // postcondition: returns array of colleges in region
        //                  where low <= tuition <= high;
        //                  the size of the array returned is equal to the
        //                  number of colleges that meet the criteria

    private:

        apvector<College> myColleges;
        // myColleges.length() is the number of colleges

    // ... other private data members not shown
};
```

## 2003 AP<sup>®</sup> COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

The following chart shows an example of colleges that could appear in an object of type `CollegeGroup`.

	<b>Name</b>	<b>Region</b>	<b>Tuition</b>
<b>0</b>	Colgate University	Northeast	\$27,025
<b>1</b>	Duke University	Southeast	\$26,000
<b>2</b>	Kalamazoo College	Midwest	\$19,764
<b>3</b>	Stanford University	West	\$25,917
<b>4</b>	Florida International University	Southeast	\$10,800
<b>5</b>	Dartmouth College	Northeast	\$27,764
<b>6</b>	Spelman College	Southeast	\$11,455

- (a) Write the `CollegeGroup` member function `UpdateTuition`, which is described as follows.  
`UpdateTuition` changes the tuition of the college whose name is passed as a parameter.

For example, if the object `colleges` is of type `CollegeGroup` and contains the entries shown in the chart above, the call `colleges.UpdateTuition("Colgate University", 27500)` would change the tuition of Colgate University to \$27,500.

Complete function `UpdateTuition` below.

```
void CollegeGroup::UpdateTuition(const apstring & collegeName,
                                int newTuition)
// precondition: collegeName exists in this CollegeGroup
// postcondition: the tuition for collegeName is changed to newTuition
```

## 2003 AP<sup>®</sup> COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

(b) The table below is repeated for your convenience.

	Name	Region	Tuition
0	Colgate University	Northeast	\$27,025
1	Duke University	Southeast	\$26,000
2	Kalamazoo College	Midwest	\$19,764
3	Stanford University	West	\$25,917
4	Florida International University	Southeast	\$10,800
5	Dartmouth College	Northeast	\$27,764
6	Spelman College	Southeast	\$11,455

Write the `CollegeGroup` member function `GetCollegeList`, which is described as follows. `GetCollegeList` returns an array of colleges that are located in the specified region and whose tuition is in the range between `low` and `high`, inclusive. The size of the array should be equal to the number of colleges that meet the criteria of region and tuition range.

For example, if the object `colleges` is of type `CollegeGroup` and contains the entries shown in the chart above, the call `colleges.GetCollegeList("Southeast", 10000, 20000)` should return an array of size two containing Florida International University and Spelman College (note that Duke University is not included because its tuition is not in the specified range and Kalamazoo College is not included because it is not in the specified region).

Complete function `GetCollegeList` below.

```
apvector<College> CollegeGroup::GetCollegeList(const apstring & region,
                                              int low, int high) const
// precondition: low <= high
// postcondition: returns array of colleges in region
//                where low <= tuition <= high;
//                the size of the array returned is equal to the number
//                of colleges that meet the criteria
```

## 2003 AP<sup>®</sup> COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

2. Periodically, a company processes the retirement of some of its employees. In this question, you will write functions to help the company determine whether an employee is eligible to retire and to process the retirement of all eligible employees.

The `Employee` class is declared as follows.

```
class Employee
{
    public:
        int Age() const;
        // returns the age (in years) of this employee

        int YearsOnJob() const;
        // returns the number of years this employee has worked

        double Salary() const;
        // returns the salary of this employee in dollars

        int ID() const;
        // returns unique employee ID number

    // ... constructors, other member functions and data not shown
};
```

The `Company` class is declared as follows.

```
class Company
{
    public:
        void ProcessRetirements();
        // postcondition: all retirement-eligible employees have been
        //                  removed from empList; empList has been resized
        //                  to reflect retirements;
        //                  empList remains sorted by employee ID;
        //                  salaryBudget has been updated to reflect remaining
        //                  employees

        // ... constructor and other public methods not shown

    private:
        bool EmployeeIsEligible(const Employee & emp) const;
        // postcondition: returns true if emp is eligible to retire;
        //                  otherwise, returns false

        apvector<Employee> empList;
        // empList.length() is the number of employees in this company

        int retireAge;           // minimum age to retire
        int retireYears;         // minimum years on job to retire
        double retireSalary;     // minimum salary to retire

        double salaryBudget;
        // total salary of all employees
};
```

**AP<sup>®</sup> COMPUTER SCIENCE A  
2003 SCORING GUIDELINES**

**Question 1**

<b>Part A:</b>	<code>UpdateTuition</code>	<b>3 pts</b>
----------------	----------------------------	--------------

- +1    loop over `myColleges`
  - +1/2    attempt
  - +1/2    correct (Note: OK to exit after a match is found)
  
- +1    identify college in array, based on `collegeName`
  - +1/2    attempt
  - +1/2    correct (must be in context of loop)
  
- +1    set tuition
  - +1/2    attempt (must call `myColleges[k].SetTuition(...)` possibly with bad syntax on array or error in parameters. `SetTuition(myCollege...)` does not get attempt)
  - +1/2    correct

<b>Part B:</b>	<code>GetCollegeList</code>	<b>6 pts</b>
----------------	-----------------------------	--------------

- +1/2    correctly declare result vector  
          (empty list or numeric constant size without subsequent `resize` loses this pt)
  
- +1/2    initialize counter for colleges added to result vector
  
- +1    loop over `myColleges`
  - +1/2    attempt (must have loop that attempts to add items to result vector)
  - +1/2    correct
  
- +1 1/2    identify college to add to vector
  - +1/2    attempt (must compare some parameter with data from `myColleges`, possibly incorrectly)
  - +1    correct
  
- +1    add identified college to result vector
  
- +1/2    increment counter for matching colleges (must use correctly when inserting into result vector)
  
- +1/2    final size of result vector is correct (must be in context of attempt at loop over `myColleges`)  
          (must also be sufficiently large throughout)
  
- +1/2    return result vector (must be in context of attempt at loop over `myColleges`)

**Usage: -1**    `CollegeGroup[k]` instead of `myColleges[k]`