AP® Computer Science A
2003 Sample Student Responses

For the College Board's online home for AP professionals, visit AP Central at apcentral.collegeboard.com.

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

	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

(a) Write the `College` 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
{
    int x;

    for (x=0; x < myColleges.length(); x++)
    {
        if (collegeName == myColleges[x].Name())
            myColleges[x].SetTuition(newTuition);
    }
}
```

Part (b) begins on page 6.

GO ON TO THE NEXT PAGE.

-5-

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
```

```
{
  int X;
  apvector<college> list(0);
  int size = 0;
  for (x = 0; X < myColleges.length(); x++)
  {
    if ((myColleges[X].Region == region) && (myColleges[X].Tuition >= low) &&
        (myColleges[X].Tuition <= high))
    {
      size++;
      list.resize(size);
      list[size-1] = mycolleges[X];
    }
  }
  return (list);
}
```

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

	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

(a) Write the `College` 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
```

```
int k;
for(k = 0; k < myColleges.length(); k++)
{
    if (myColleges[k].Name() == collegeName)
        myColleges[k].SetTuition(newTuition);
}
```

Part (b) begins on page 6.

GO ON TO THE NEXT PAGE.

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
```

```
apvector <College> Temp ( myColleges.length() );
int count = 0;
int k;
for (k=0; k< myColleges.length())
{
    if ( myColleges[k].Region() == region )
    {
        if ( myColleges[K].Tuition() >= low )
        {
            if ( myColleges[k].Tuition() <= high )
            {
                Temp[count] = myColleges[k];
                count ++;
            }
        }
    }
}
Temp.resize(count);
return Temp;
```

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

	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

(a) Write the `College` 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
{
    for (int index = 0; index < myColleges.length(); index++)
    {
        if (myColleges[index].Name() == collegeName)
        {
            myColleges[index].SetTuition(newTuition);
        }
    }
    return;
}
```

Part (b) begins on page 6.

GO ON TO THE NEXT PAGE.

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
```

```
{
    apvector <college> List;
    for (int index = 0; index < myColleges.length(); index++)
    {
        if ((myColleges[index].Region() == region) && ((myColleges[index].Tuition() <=
        low) && (myColleges[index].Tuition() >= high)))
        {
            List = List + myColleges.Name();
        }
    }

    return List;

}
```