NAME: ________________________________

```
#include <cassert>

/*
+-----------------------------------------------------------------+
|                                                                 |
|                           SchoolBus                             |
|                                                                 |
+-----------------------------------------------------------------+
|                                                                 |
| - seats : int                                                   |
| - seatedStudents : int                                          |
|                                                                 |
+-----------------------------------------------------------------+
|                                                                 |
| + SchoolBus(seats : int, seatedStudents : int)                  |
| + addStudents(students : int) : bool                            |
| + removeStudents(students : int) : bool                         |
| + getStudents() : int                                           |
|                                                                 |
+-----------------------------------------------------------------+

Instances of the SchoolBus class represent school buses of varying
sizes. The class provides functions to add and remove students.
The add and remove functions return false and do not modify the state
of the object if the operation can not be done exactly as requested.
For example, if there are 3 empty seats and you try to add 5
students, then the addStudents function returns false and does not
change the number of seated students. There is a single accessor
function called getSeatedStudents, which returns the total number of
students seated in the bus.
*/
```

Figure 1

```
class SchoolBus
{
public:
    SchoolBus(int seats, int seatedStudents);
    bool addStudents(int students);
    bool removeStudents(int students);
    int getStudents();

private:
    int seats;
    int seatedStudents;
};
```

Figure 2

INSTRUCTIONS

Figure 1 contains the UML class diagram and description for a class called SchoolBus. Figure 2 contains a declaration of the SchoolBus class. All of the questions in this quiz refer to the SchoolBus class as described in Figures 1 and 2.

1) Provide an implementation of the SchoolBus constructor. (25 points)

2) Provide an implementation of the getStudents function. (25 points)

3) Provide an implementation of the addStudents function. (25 points)

4) Provide an implementation of the removeStudents function. (25 points)

5) Provide test code for the addStudents function you implemented in problem 3. Use assert statements for this purpose. Make sure the test code executes all lines of code in the function. (25 points)

6) Provide an alternative implementation of the addStudents function that sets the number of students seated in the bus to its maximum when the added students exceed the available capacity. The function should continue to return false in this case. (25 points)

7) Provide an alternative implementation of the removeStudents function that sets the number of students seated on the bus to zero when asked to remove more students than are already seated. The function should continue to return false in this case. (25 points)

8) Provide test code for the removeStudents function you implemented in problem 7. Use assert statements for this purpose. Make sure the test code executes all lines of code in the function. (25 points)