This program will not compile/build until the Student class (Tests 1 and 2) is completed.

Test 1 – Member fields and Constructors

```
Student Test1(string last, string first, int idNo)
```

For this test you will add member fields and constructors to the Student class.

The private member fields to add are a string to represent a last name, a string to represent a first name and an int to represent an identification number.

You will provide the default (no parameter) constructor and one overloaded constructor.

- The default constructor will set last and first names to empty strings ("") and set the id number to 1000000 (1 million).
- The overloaded constructor that will accept two strings (last name, first name) and one int (id number) and update the member fields the received values.

After making the specified additions in the Student class, create and then return a new Student object using the provided parameters.

Test 2 – Getters and Setters

```
Student Test2()
```

For this test, you are to add public getters and setters for the Student class member fields. The methods must be named:

Getters	**Setters**
GetFirstName	SetFirstName
GetLastName	SetLastName
GetIDNumber	SetIDNumber

After making the specified additions in the Student class, create and then return a new Student object **using the default constructor**.

For Tests 3-5, you will use the enrollment array. It is a static member of the Submission class.

Test 3 – Insert in array

```
bool Test3(Student enrolled)
```

Given a string, `enrolled`, search the `enrollment` array to find an empty element (null). If an empty element is located, place `enrolled` in the empty element and return `true`. If there are no available slots, return `false`.

Test 4 – Remove from array

```
bool Test4(int idNumber)
```

Given an int, `idNumber`, search the array to find `idNumber`. Examine each element in the `enrollment` array to find `idNumber`. If it is found, mark the array location as empty (`null`) and return `true`. If `idNumber` is not found in the array, return `false`.

Test 5 – Retrieve from array

```
Student Test5(int idNumber)
```

Given an int, `idNumber`, search the array to find `idNumber`. Examine each element in the `enrollment` array to find `idNumber`. If it is found, return the Student with the id number `idNumber`. If `idNumber` is not found in the array, return `null`.

Notes:

- Identifiers (method names and variables) are case sensitive. For example, SetIdNumber is not the same as SetIDNumber. If a specific name is provided, you must use the exact name provided.
- For Tests 4 and 5, before attempting to access a Student class method from an array element, you must ensure the array element contains a Student object. If the element is null and you attempt to call a member method, the program will throw an Exception and crash. To verify the element is not empty, compare the element to null – if the element is not null, then it contains a valid Student object.