Chapter 3: Part-C
Classes, Objects, and Basic Structural Modeling in UML

H.S. Sarjoughian

CSE 460: Software Analysis and Design

School of Computing, Informatics and Decision Systems Engineering
Fulton Schools of Engineering

Arizona State University, Tempe, AZ, USA

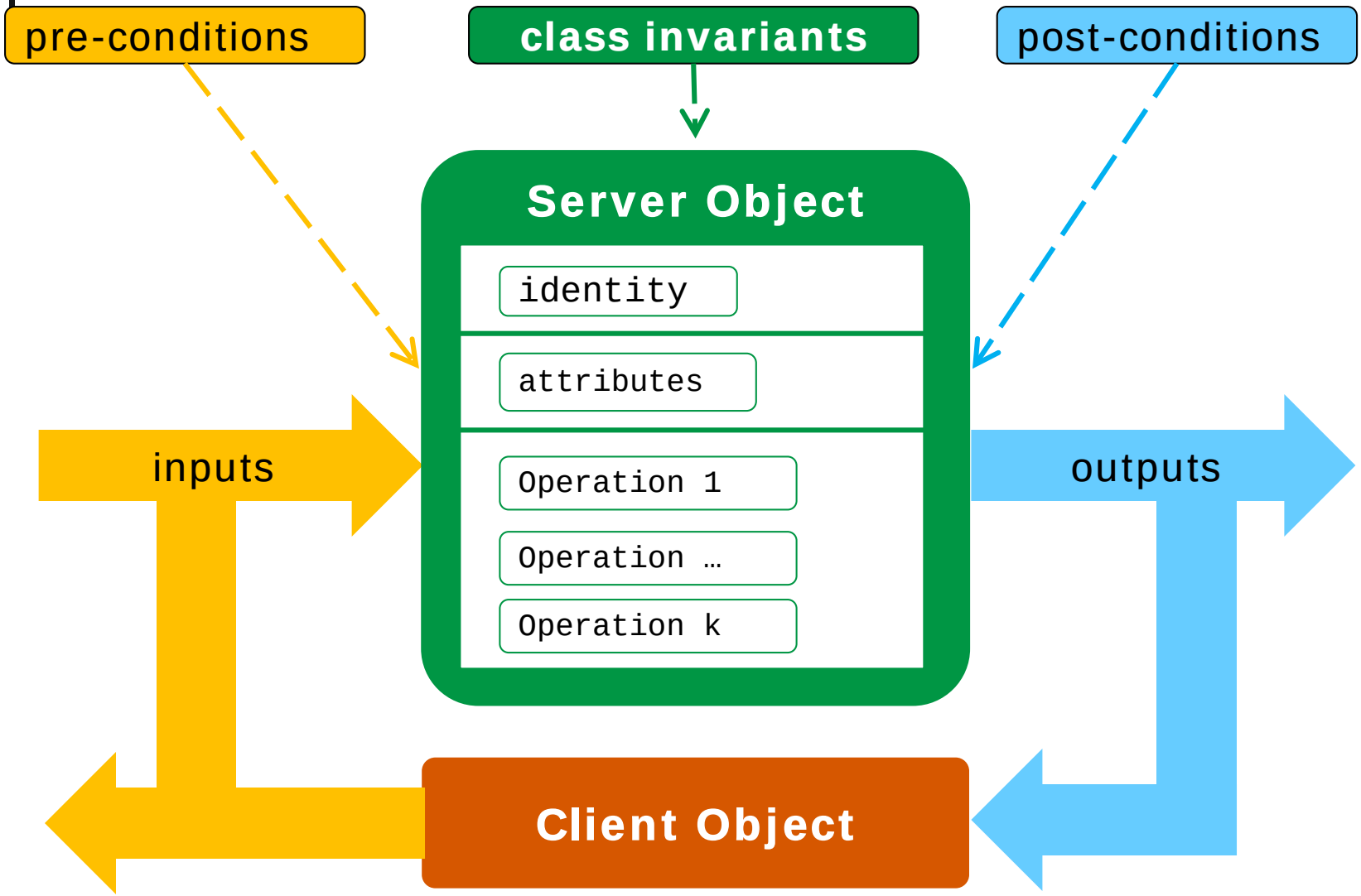
Design-by-Contract

- Design-by-Contract is fundamental for developing high-quality software
  - Characterize behavior of objects of a class during their lifetime (i.e., creation, operations, and destruction)
  - Operations on objects can change their states
    - Understand the effect of a sequence of operations applied to an object
- Class Invariant
  - Characterizes object states – i.e., constrains the state stored in the instances of a class
  - A condition (logical assertion) that is satisfied during the execution
  - Established during construction of objects and constantly preserved.

Design-by-Contract (cont.)

- Loop invariant
  - An ***assertion*** which must be satisfied prior to the first execution of a loop, and preserved by every iteration, so that it will hold on loop termination.
- Pre-condition
  - An ***assertion*** attached to an *operation*, which must be guaranteed by every ***client*** prior to any call to the operation.
- Post-condition
  - An ***assertion*** attached to an *operation*, which must be guaranteed by the operation's body on return from any call to the operation if the ***pre-condition*** was satisfied on entry.

Design-by-Contract



A Simple Example

- Consider class **`Impl`** as an array of size `length` implementing abstract class **`Set`**.
  - `delete` operation for a variable x

- ***class invariant***

```
for all i, j (1 ≤ i ≤ length and 1 ≤ j ≤ length and i≠j)
implies impl[i] ≠ impl[j]
```

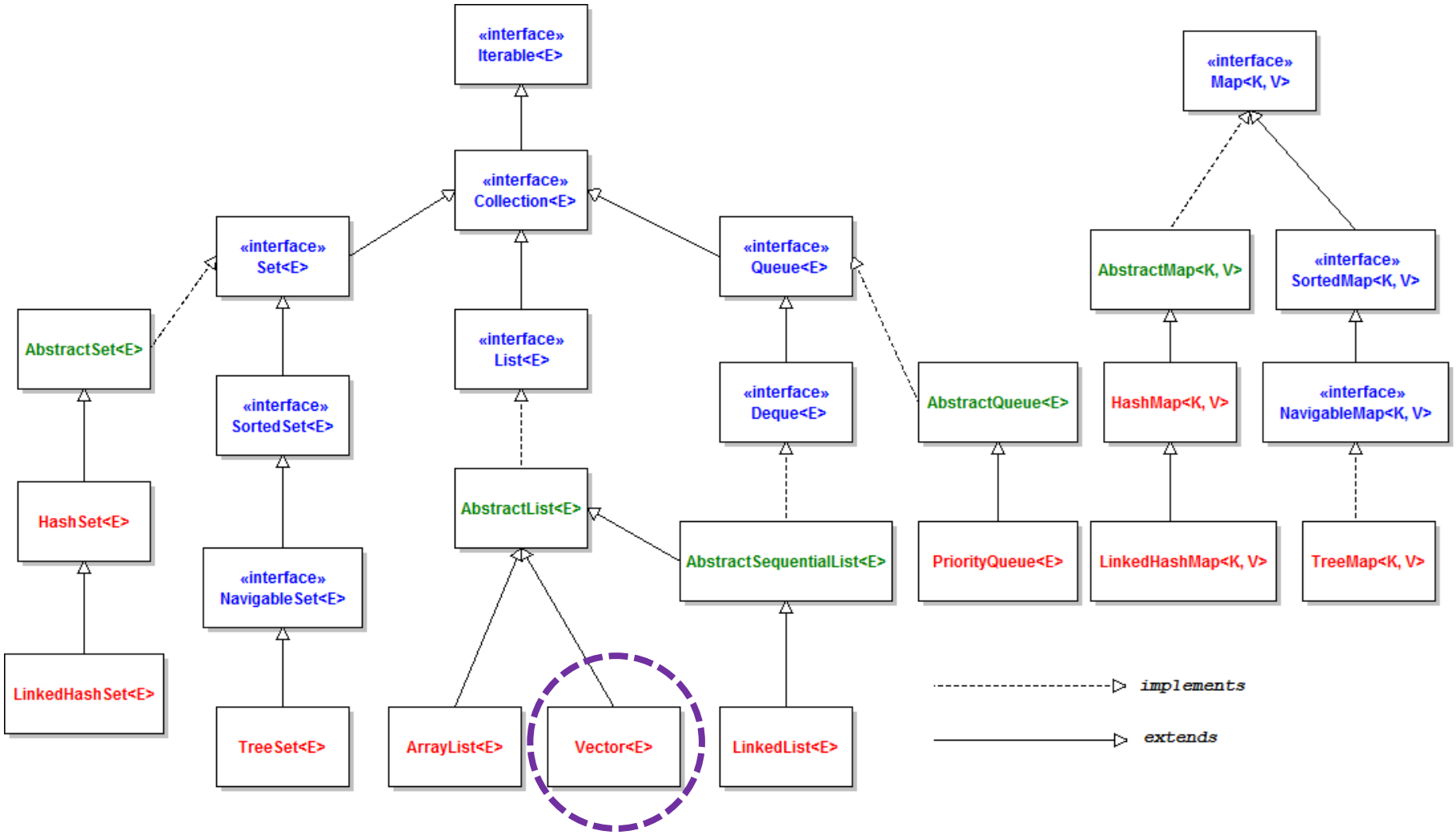
- ***pre-condition***

```
exists i (1 ≤ i ≤ length and impl[i] = x)
```

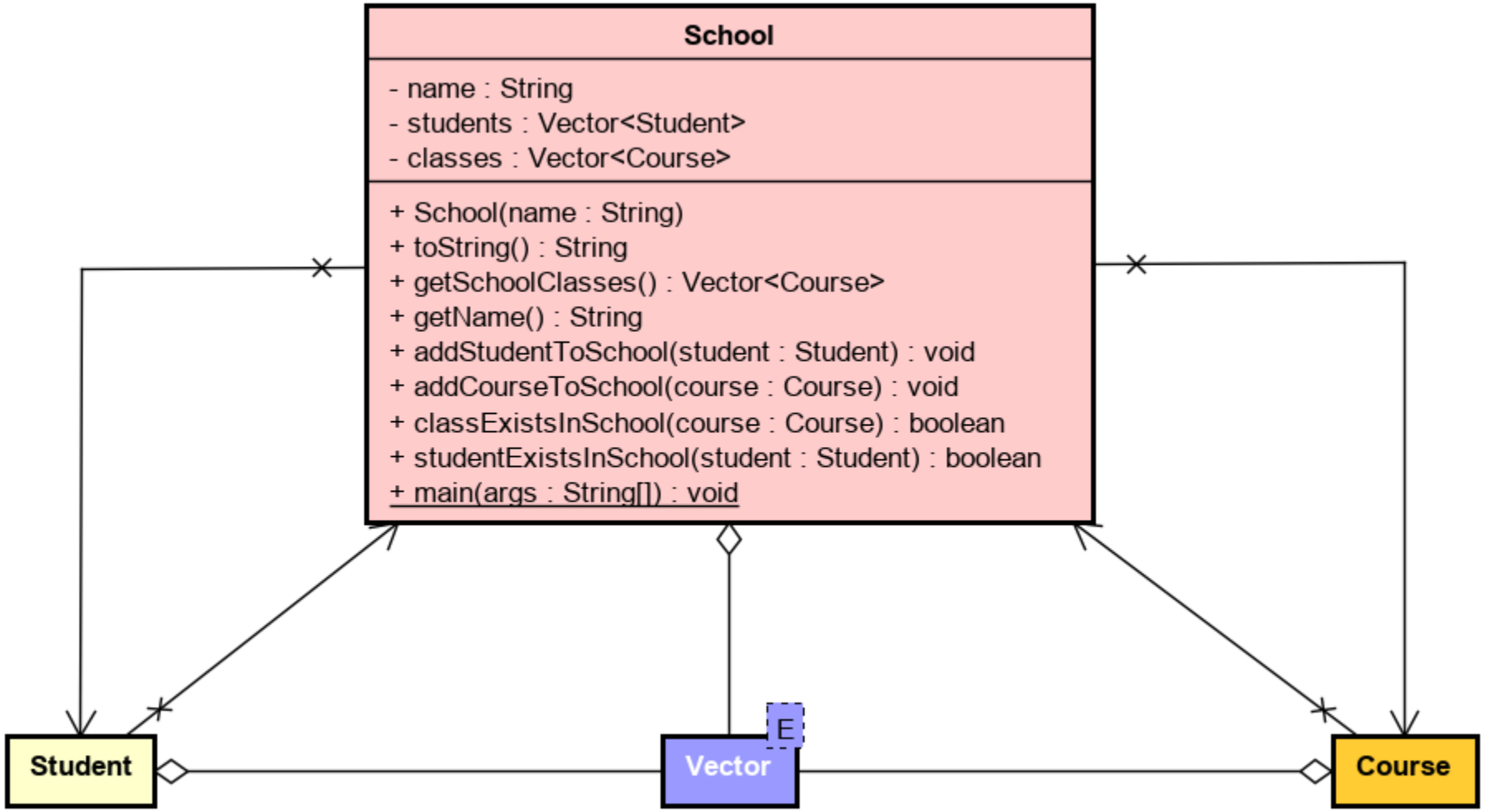
- ***post-condition***

```
for all i (1 ≤ i ≤ length implies impl[i] ≠ x) and
for all i ((1 ≤ i ≤ old-length and old-impl[i] ≠x) implies
   exists j (1 ≤ i ≤ length and impl[i] = old-impl[i]))
```

Java Collection API



Course Registration Example



A Method in the Course Class

```
public static void createClassRosterArray(Course course,
    Student[] classRosterArray) {
/*
    * copyInto(Object[] anArray)
    * copies the components of this vector into the specified array.
    **/

    course.classRoster.copyInto(classRosterArray);

    for(int i = 0; i < classRosterArray.length; i++) {
        System.out.println(classRosterArray[i]);
    }
}
```

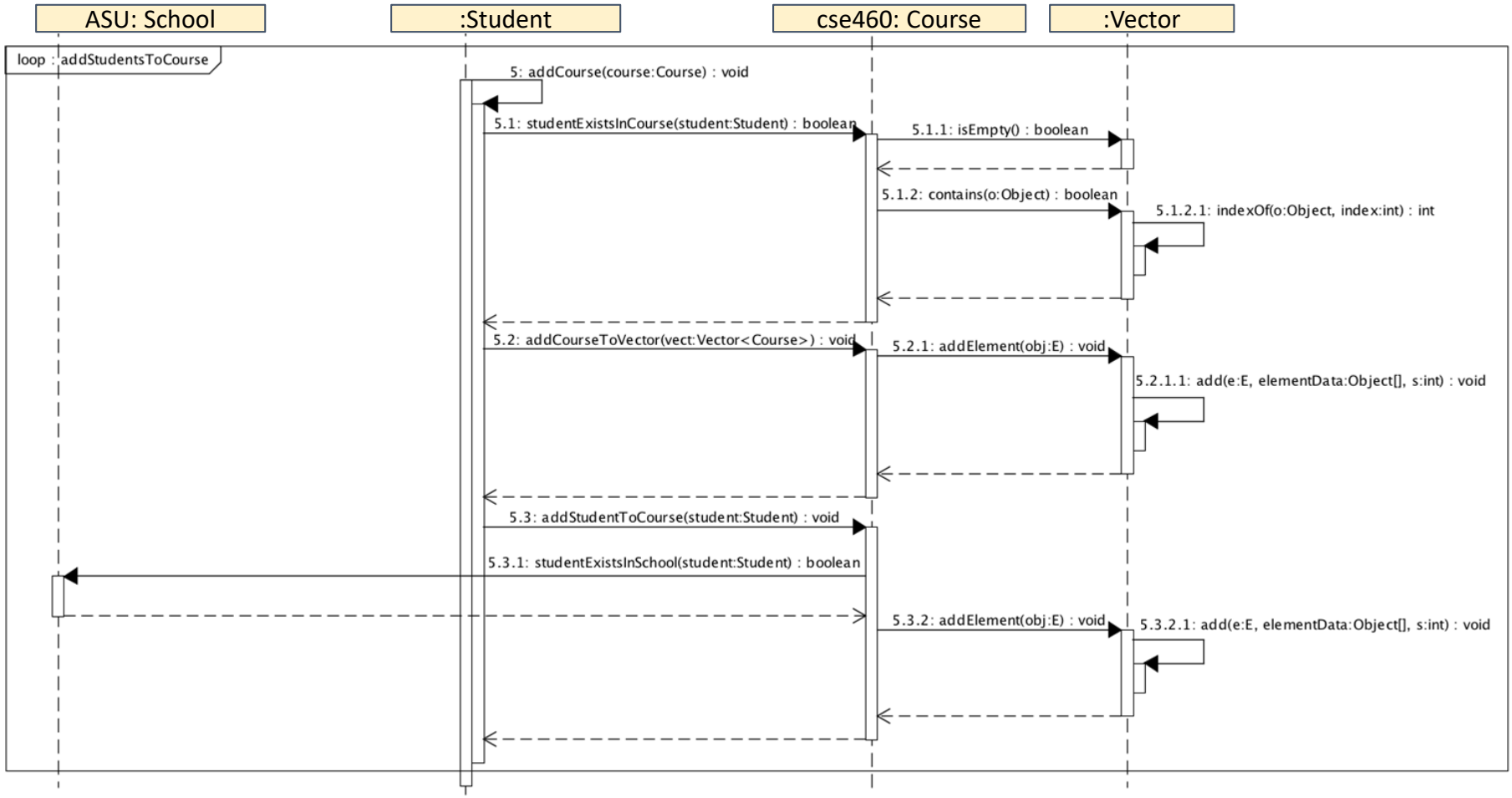
A Method in the Course Class

```
public boolean studentExistsInCourse(Student student) {
    /*
    * isEmpty()
    * Tests if this vector has no components
    **/

    if (this.classRoster.isEmpty()) return false;
    /*
    * contains(Object o)
    * Checks if the vector contains the specified element
    **/

    return this.classRoster.contains(student);
}
```

A Partial Sequence Diagram



Observations

- Design by Contract is key for developing high quality software
  - Helps to reduce design bugs and errors
  - Makes software more reliable
- The Object Constraint Language is a useful way to implement invariants on a UML model
- Class invariants and loop invariants are important parts of Design by Contract