

## Homework 1

4.2. Below is what myList looks like after each operation

Operation	Data	Position
myList.add("alpha");	alpha	1
myList.add(1, "beta");	beta	1
	alpha	2
myList.add("gamma");	beta	1
	alpha	2
	gamma	3
myList.add(2, "delta");	beta	1
	delta	2
	alpha	3
	gamma	4
myList.add(4, "alpha");	beta	1
	delta	2
	alpha	3
	alpha	4
	gamma	5
myList.remove(2);	beta	1
	alpha	2
	alpha	3
	gamma	4
myList.remove(2);	beta	1
	alpha	2
	gamma	3
myList.replace(3, "delta");	beta	1
	alpha	2
	delta	3

After all of the above operations are executed, myList will have (beta, 1) → (alpha, 2) → (delta, 3)

4.3. *Task: Searches for an entry in the list.*  
*\* @param anObject entry to be searched*  
*\* @return position represented as an integer of the*  
*\* first occurrence of the entry,*  
*\* or -1 if not found \*/*  
**public int** getPosition(T anObject);

4.4. *Task: Removes the first occurrence of a given*  
*\* object from the list.*  
*\* @param anObject object to be removed*  
*\* @return true if the object was found and removed,*  
*\* or false if the object was not found \*/*  
**public boolean** remove(T anObject)

4.6. **public static void** main(String [] args)  
 {  
     *// create two Car objects to be inserted in list*

```

Car c1 = new Car("Toyota", "Corolla", 1986);
Car c2 = new Car("Toyota", "Sienna", 2002);

// declare List that stores Car objects
ListInterface<Car> myList = new AList<Car>();

// insert Car objects into list
myList.add(c1);
myList.add(c2);
myList.add(1, c2);

// grab position of the first occurrence of c2
int position = myList.getPosition(c2);

System.out.println("The position of the first occurrence of c2 is: " +
    position);
}

The position of the first occurrence of c2 is: 1
4.7. // remove object at givenPosition in nameList
nameList.remove(givenPosition);

// add object newObject to givenPosition in nameList
nameList.add(givenPosition, newObject);

4.8. // used as flag for found or not
boolean nameFound = false;

int i;

// iterate through nameList
for(i = 1; i <= nameList.getLength(); i++)
{
    Name n = nameList.getEntry(i);

    // check if entry in nameList matches myName
    if(n.getFirst() == myName.getFirst() &&
        n.getLast() == myName.getLast())
    {
        nameFound = true;
        break;
    }
}

// if found, print position in nameList
if(nameFound)
{
    System.out.println("The Name object, myName, was found in position " +
        i + ".");
}

4.9. (a) // iterate through studentList
for(int i = 1; i <= studentList.getLength(); i++)
{
    // create reference to entry
    Student s = studentList.getEntry(i);

```

```

    // create reference to full name of entry
    Name n = s.getName();

    // extract and print out last name from reference to full name
    System.out.println(n.getLast());
}

```

```

(b) // store current first student
    Student oldFirst = studentList.getEntry(1);

    // replace first student with last student
    studentList.replace(1, studentList.getEntry(studentList.getLength()));

    // replace last student with previous first student
    studentList.replace(studentList.getLength(), oldFirst);

```

```

4.10. (a) // check if quizScores is empty
    if(!quizScores.isEmpty())
    {
        // set initial lowest score to be the first one
        double lowest = quizScores.getEntry(1);

        int position = 1;

        // iterate through quizScores
        for(int i = 2; i <= quizScores.getLength(); i++)
        {
            // compare if current score is lower than the previous score
            if(quizScores.getEntry(i) < lowest)
            {
                // if so, choose current score over previous one
                lowest = quizScores.getEntry(i);
                position = i;
            }
        }

        // remove lowest score from quizScores
        quizScores.remove(position);
    }

(b) if(!quizScores.isEmpty())
{
    double sum = 0;

    // iterate through quizScores
    for(int i = 1; i <= quizScores.getLength(); i++)
    {
        // sum up all entries
        sum += quizScores.getEntry(i);
    }

    // calculate average from sum and length of quizScores
    double average = sum/(double)(quizScores.getLength());
    System.out.println("The average of the scores in quizScores is: "
        + average);
}

```

```
4.11. int i = 1;

while(i <= coinList.getLength())
{
    coinList.toss(i);

    if(coinList.isHeads(i))
    {
        // if heads move entry from coinList to headsList
        headsList.add(coinList.remove(i));
    }
    else
    {
        // increment only if we don't move entry
        i++;
    }
}

System.out.println("Total heads that came up: "
+ headsList.getLength());
```