**Gebze Technical University**
**Computer Engineering**

**CSE 222 - 2018 Spring**

# HOMEWORK 2 REPORT

**SEVGİ BAYANSALDUZ**
**151044076**

MEHMET BURAK KOCA

## 1 INTRODUCTION

### 1.1 Problem Definition

- **Part 1**

  Creating a course structure using the Linked List class. This structure has 3 methods;

  **getByCode** :Returns all courses according  to given course code.

  **listSemesterCourses** ; Returns all courses according to given semester.

  **getByRange**: Returns all courses from given start index to last index

- . **Part 2**

  Make an expanded Java LİnkedLİst structure.This structure must has **enable**(), **disable**() and **showDiabled**() methods.

- **Part 3**

  Implement a new course list structure. Courses are linked as a list form int this structure also  courses in same semester are linked together as circular list

### 1.2 System Requirements

- **Part 1**

  The GTUCoursesList class reads the Courses(CSV)(Updated).csv file with the Read Class method ,when creating the object. Therefore the name of the file should not be changed and must be in the folder.

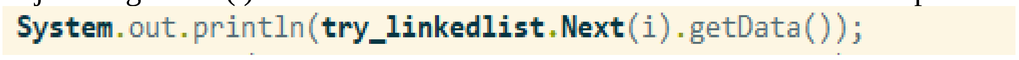
- **Part 2**

  The Part2LinkedList class methods requirements:
  **Disable**: Disable takes Object as a parameter.
  **Enable**: Enable takes Object as  a parameter
  **ShowDiasbled**: This method does not take any parameter.

- **Part 3**

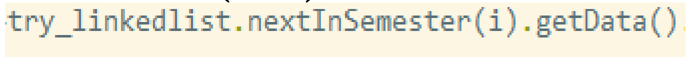
  The GTULinkList class methods requirements:

  **add(data):** The add method, which takes only one parameter, takes the GTUCourse object as a parameter.
  **add(index,data):** The add method, which takes two parameters,takes integer and GTUCourse object as a parameter.
  **remove(index):**  The remove method takes integer as a parameter.
  **size():**  This method does not takes any parameter and return type is integer.
  **Next(index):** This method takes integer aa a parameter. Given parameter points the index  and the method returns the next node of this index.**Return** type is NodeCourse object so getData( ) function is used to return the next course.For example:



```
System.out.println(try_linkedlist.Next(i).getData());
```

  **NextInSemester(index):**This methods works like **Next(index).** Example



```
try_linkedlist.nextInSemester(i).getData()
```

# 2 METHOD

## 2.1 Class Diagrams

- **Part1**

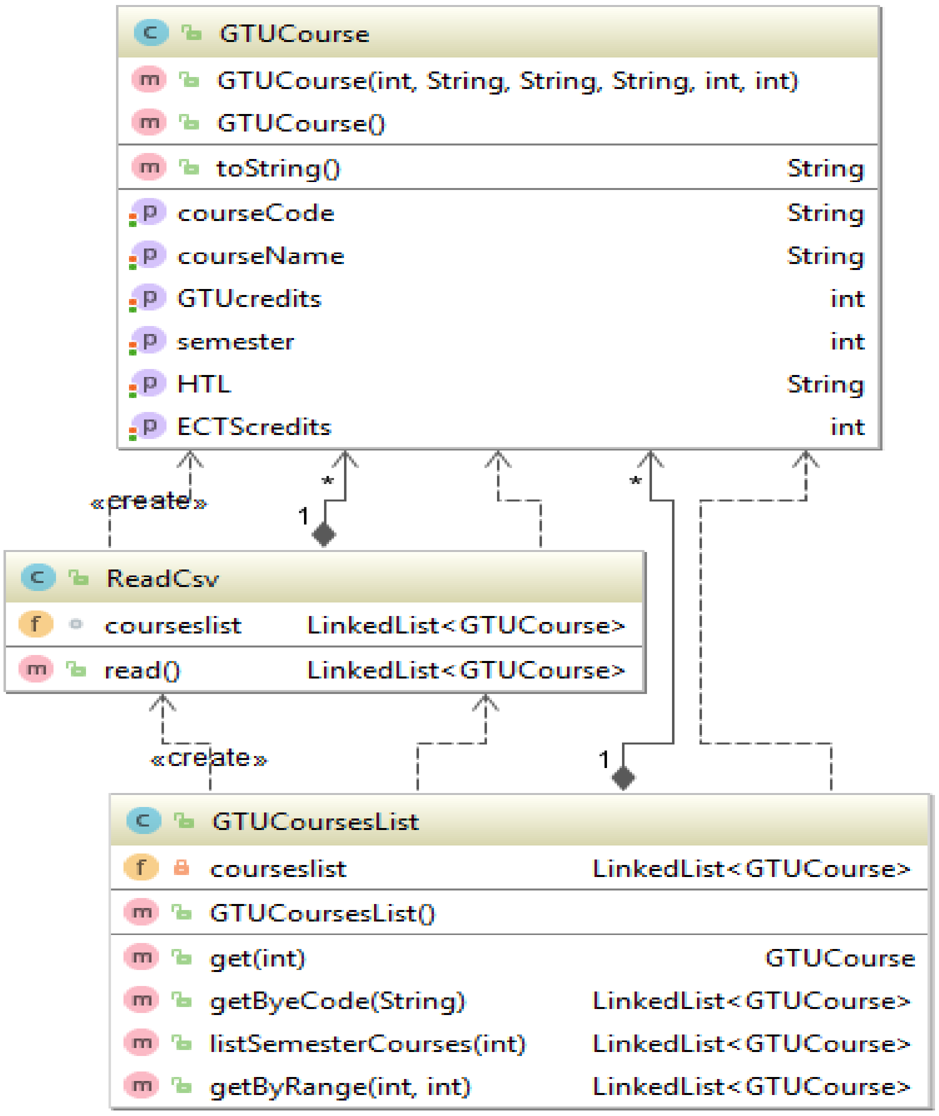

GTUCourse
GTUCourse(int, String, String, String, int, int)
GTUCourse()
toString() String
courseCode String
courseName String
GTUcredits int
semester int
HTL String
ECTScredits int
«create»
*
1
*
ReadCsv
courseslist LinkedList<GTUCourse>
read() LinkedList<GTUCourse>
«create»
1
GTUCoursesList
courseslist LinkedList<GTUCourse>
GTUCoursesList()
get(int) GTUCourse
getByeCode(String) LinkedList<GTUCourse>
listSemesterCourses(int) LinkedList<GTUCourse>
getByRange(int, int) LinkedList<GTUCourse>

**PHOTO 1.1**

ReadCsv class creates list of courses with read() method and returns LinkedList<GTUCourse>. GTUCoursesList constructor calls read method and and assign this method type to the courseslist data field.

- **PART 2**

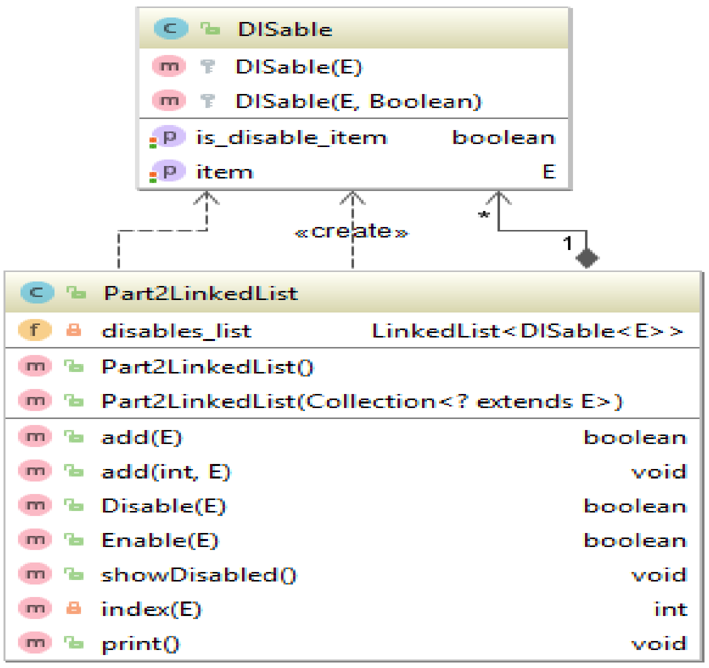


**PHOTO 1. 2**

Part2LinkedList class creates collection which type is DISable object.DISable class stores item and item`s information. If is_disable_item is true,that means item is disable,otherwise item is enable.

- **Part 3**

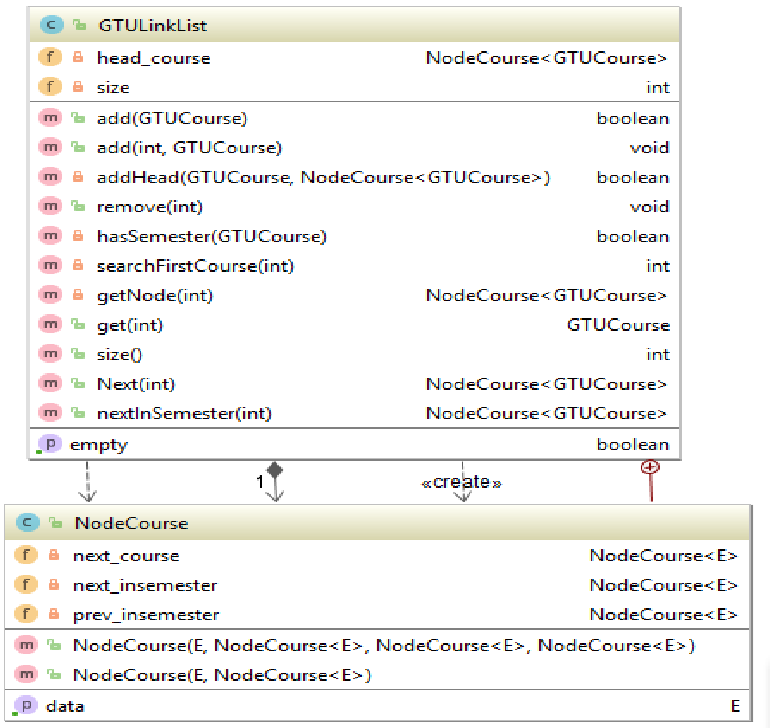

GTULinkList
head_course NodeCourse<GTUCourse>
size int
add(GTUCourse) boolean
add(int, GTUCourse) void
addHead(GTUCourse, NodeCourse<GTUCourse>) boolean
remove(int) void
hasSemester(GTUCourse) boolean
searchFirstCourse(int) int
getNode(int) NodeCourse<GTUCourse>
get(int) GTUCourse
size() int
Next(int) NodeCourse<GTUCourse>
nextInSemester(int) NodeCourse<GTUCourse>
empty boolean
1
«create»
NodeCourse
next_course NodeCourse<E>
next_insemester NodeCourse<E>
prev_insemester NodeCourse<E>
NodeCourse(E, NodeCourse<E>, NodeCourse<E>, NodeCourse<E>)
NodeCourse(E, NodeCourse<E>)
data E

**PHOTO 1.3**

NodeCourse class is inner class of the GTULinkList class.

### 2.2 Use Case Diagrams

Add use case diagrams if required.

### 2.3 Other Diagrams (optional)

Add other diagrams if required.

### 2.4 Problem Solution Approach

During the project development phase, the objects that should be used first were con-sidered and classes were created for those objects.

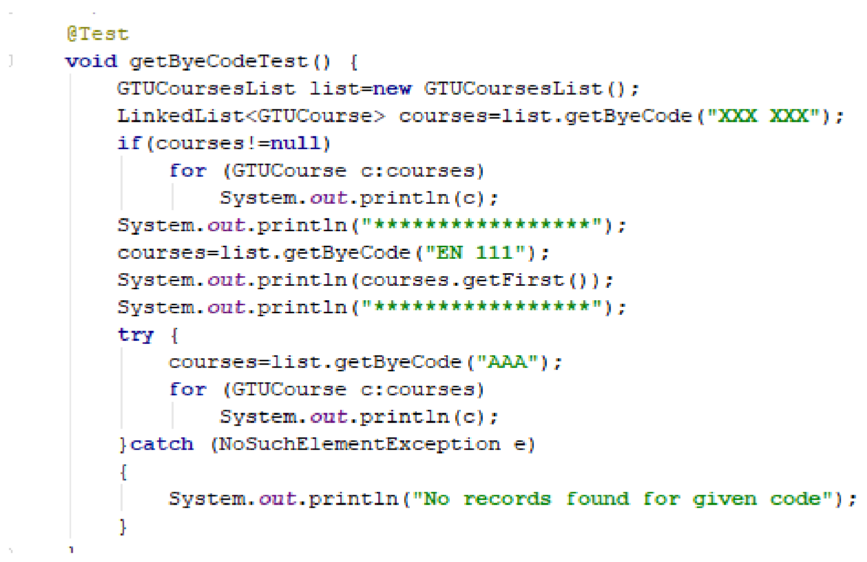
## 3 RESULT

### 3.1 Test Cases

- **PART 1**

  **1. getByCode (string code)**

  Test for getByCodeTEst() method.

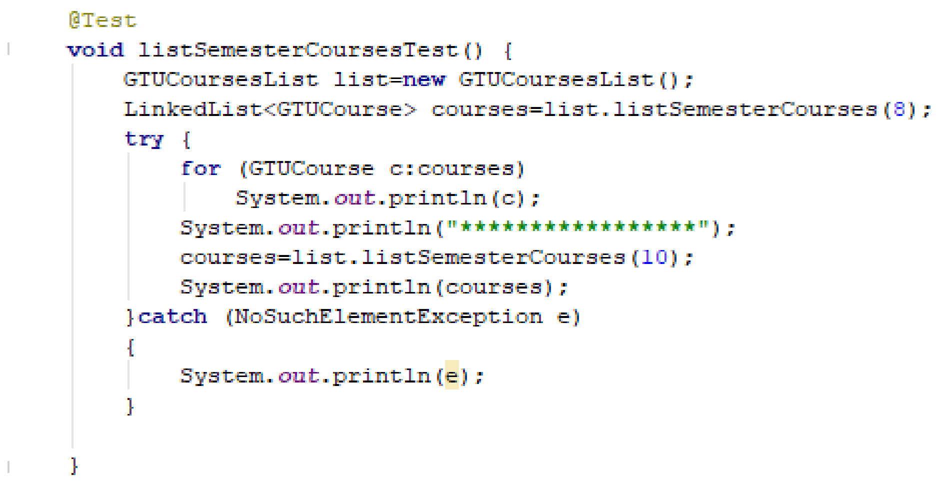


```
@Test
void getByeCodeTest() {
    GTUCoursesList list=new GTUCoursesList();
    LinkedList<GTUCourse> courses=list.getByeCode("XXX XXX");
    if(courses!=null)
        for (GTUCourse c:courses)
            System.out.println(c);
    System.out.println("****************");
    courses=list.getByeCode("EN 111");
    System.out.println(courses.getFirst());
    System.out.println("****************");
    try {
        courses=list.getByeCode("AAA");
        for (GTUCourse c:courses)
            System.out.println(c);
    }catch (NoSuchElementException e)
    {
        System.out.println("No records found for given code");
    }
}
```

**Photo 3.1.1**

Firstly, the GTUCoursesList class object is created. Then getByeCode method is called for "XXX XXX",and the value returned by this function is assigned to the value courses.After assigning,the value courses is printed. The same process is done for "AAA".

## 2. listSemesterCourses (int semester)

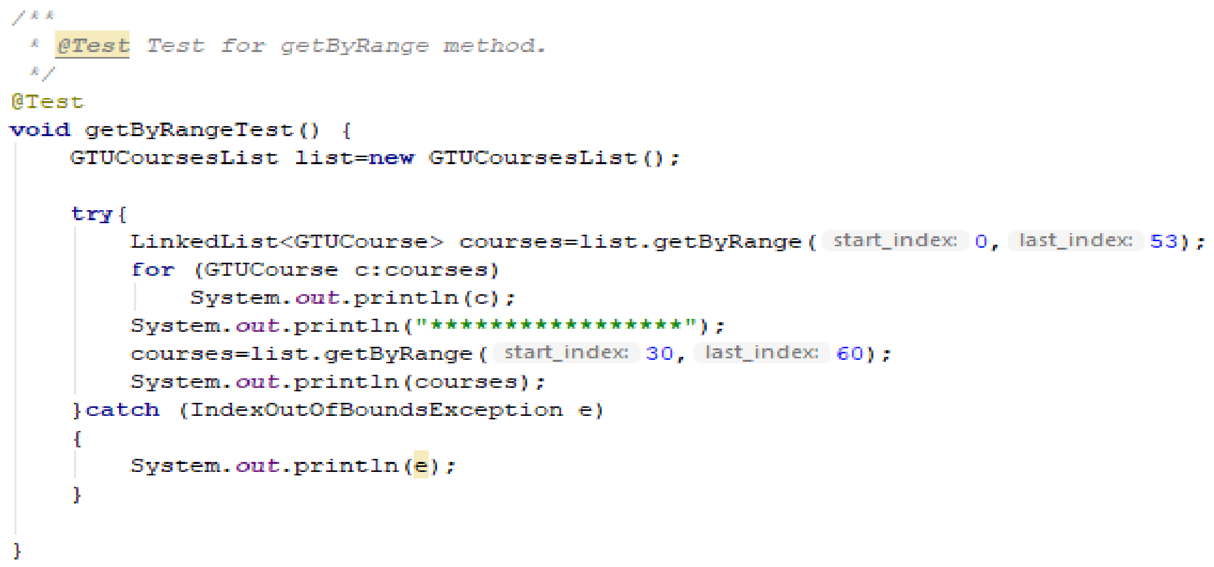


```
@Test
void listSemesterCoursesTest() {
    GTUCoursesList list=new GTUCoursesList();
    LinkedList<GTUCourse> courses=list.listSemesterCourses(8);
    try {
        for (GTUCourse c:courses)
            System.out.println(c);
        System.out.println("*****************");
        courses=list.listSemesterCourses(10);
        System.out.println(courses);
    }catch (NoSuchElementException e)
    {
        System.out.println(e);
    }

}
```

**Photo 3.1.2**

Firstly, the GTUCourselist class object is created. Then listSemesterCourses method is called for "8" and the value returned by this function is assigned to the value courses.After assigning,the value courses is printed. The same process is done for "10".

## 3. getByRange(int start_index, int last_index)



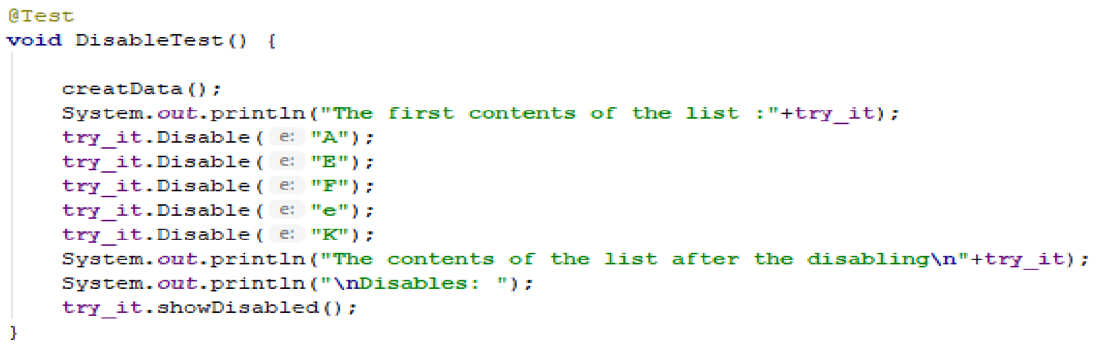
```
/**
 * @Test Test for getByRange method.
 */
@Test
void getByRangeTest() {
    GTUCoursesList list=new GTUCoursesList();

    try{
        LinkedList<GTUCourse> courses=list.getByRange( start_index: 0, last_index: 53);
        for (GTUCourse c:courses)
            System.out.println(c);
        System.out.println("*****************");
        courses=list.getByRange( start_index: 30, last_index: 60);
        System.out.println(courses);
    }catch (IndexOutOfBoundsException e)
    {
        System.out.println(e);
    }

}
```

**Photo 3.1.3**

Firstly, the GTUCourselist class object is created. Then getByRange method is called for "0 -53" and the value returned by this function is assigned to the value courses.After assigning,the value courses is printed. The same process is done for "30-60".

- **PART 2**

  1. **Disable(E e)**

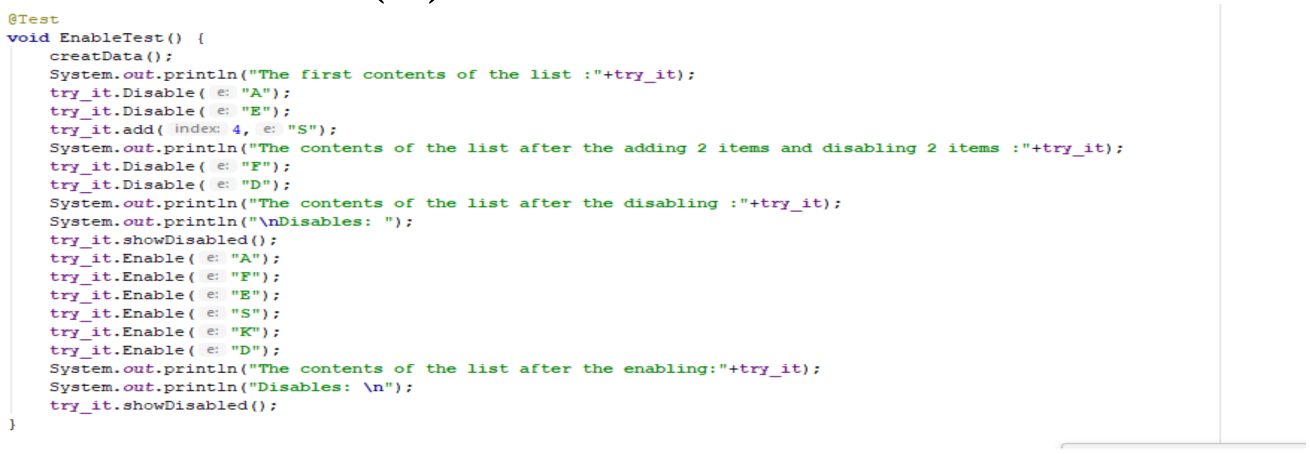


```
@Test
void DisableTest() {

    creatData();
    System.out.println("The first contents of the list :"+try_it);
    try_it.Disable( e: "A");
    try_it.Disable( e: "E");
    try_it.Disable( e: "F");
    try_it.Disable( e: "e");
    try_it.Disable( e: "K");
    System.out.println("The contents of the list after the disabling\n"+try_it);
    System.out.println("\nDisables: ");
    try_it.showDisabled();
}
```

**Photo 3.1.4**

Firstly the creatData method is called and the try_it variable is created and printed on the screen.After creating to the  data, some items of the try_it is made disable.After disabling ,the data of the try_it is printed on the screen with showDiasabled method.

  2. **Enable(E e)**



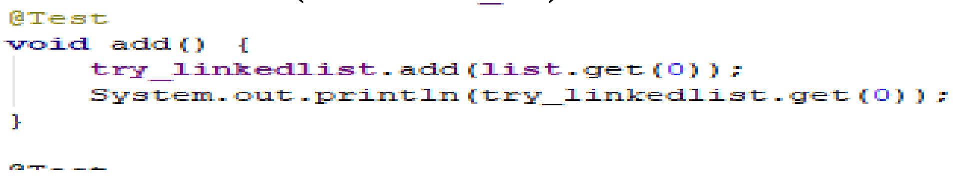
```
@Test
void EnableTest() {
    creatData();
    System.out.println("The first contents of the list :"+try_it);
    try_it.Disable( e: "A");
    try_it.Disable( e: "E");
    try_it.add( index: 4, e: "S");
    System.out.println("The contents of the list after the adding 2 items and disabling 2 items :"+try_it);
    try_it.Disable( e: "F");
    try_it.Disable( e: "D");
    System.out.println("The contents of the list after the disabling :"+try_it);
    System.out.println("\nDisables: ");
    try_it.showDisabled();
    try_it.Enable( e: "A");
    try_it.Enable( e: "F");
    try_it.Enable( e: "E");
    try_it.Enable( e: "S");
    try_it.Enable( e: "K");
    try_it.Enable( e: "D");
    System.out.println("The contents of the list after the enabling:"+try_it);
    System.out.println("Disables: \n");
    try_it.showDisabled();
}
```

**Photo 3.1.5**

Firstly the creatData method is called and the try_it variable is created and printed on the screen.After creating to the  data, some items of the try_it is made disable.After disabling ,the data of the try_it is printed on the screen with showDiasabled method.Finally some of the disabling items is made enable then enable end disable items is printed on the screen.

- **PART 3**

  1. **add(GTU Course data)**

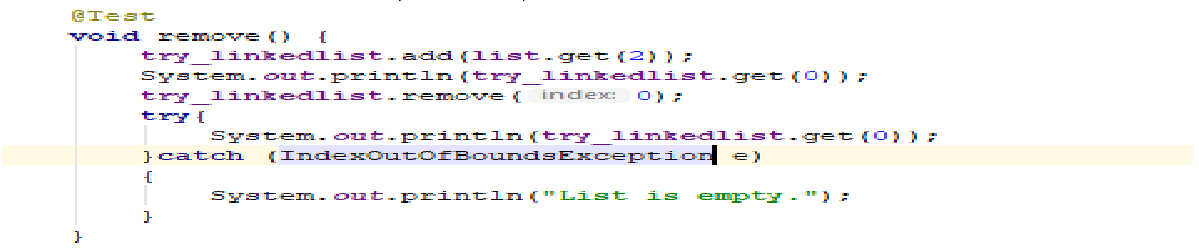


```
@Test
void add() {
    try_linkedlist.add(list.get(0));
    System.out.println(try_linkedlist.get(0));
}
```

**Photo 3.1.6**

This test class has the list and try_linkedList variables.  The first element of variable the list was added to variable linkedList

## 2. remove(int index)

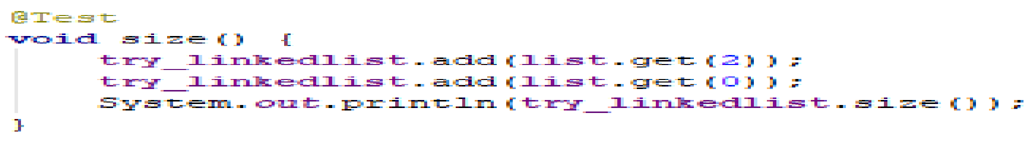


```
@Test
void remove()  {
    try_linkedlist.add(list.get(2));
    System.out.println(try_linkedlist.get(0));
    try_linkedlist.remove( index: 0);
    try{
        System.out.println(try_linkedlist.get(0));
    }catch (IndexOutOfBoundsException e)
    {
        System.out.println("List is empty.");
    }
}
```

**Photo 3.1.7**

This test class has the list and try_linkedList variables.  The first  element of variable the list is added to variable linkedList.After adding this element is removed from the variable.

## 3. Size()

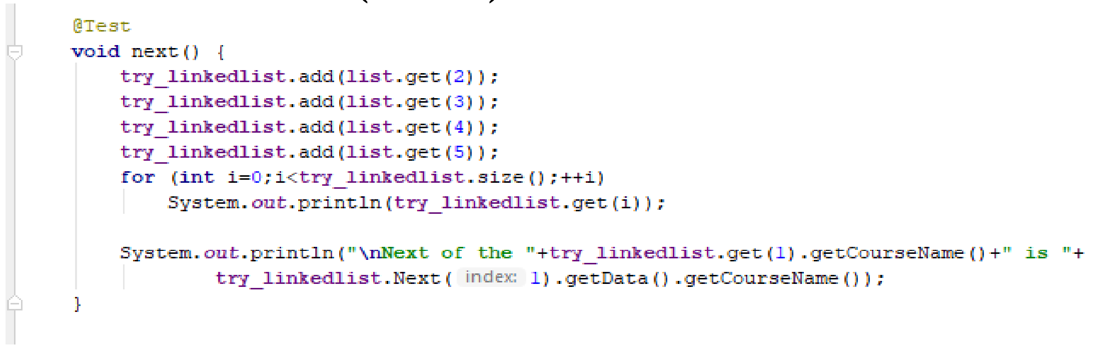


```
@Test
void size()  {
    try_linkedlist.add(list.get(2));
    try_linkedlist.add(list.get(0));
    System.out.println(try_linkedlist.size());
}
```

**Photo 3.1.8**

This test class has the list and try_linkedList variables.  The first  and third elements of variable the list is added to variable linkedList.After adding size method is called.

## 4. Next(int index)

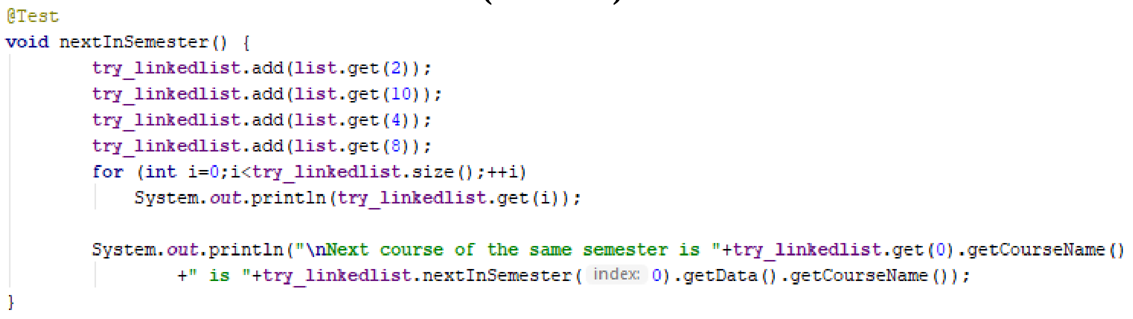


```
@Test
void next() {
    try_linkedlist.add(list.get(2));
    try_linkedlist.add(list.get(3));
    try_linkedlist.add(list.get(4));
    try_linkedlist.add(list.get(5));
    for (int i=0;i<try_linkedlist.size();++i)
        System.out.println(try_linkedlist.get(i));

    System.out.println("\nNext of the "+try_linkedlist.get(1).getCourseName()+" is "+
            try_linkedlist.Next( index: 1).getData().getCourseName());
}
```

**Photo 3.1.9**

This test class has the list and try_linkedList variables.  The elements of variable the list is added to variable linkedList. After adding data of the try_linkedList is printed on the screen.After printing, the Next() method is called for index“1”.

## 5. nextInSemester(int index)



```
@Test
void nextInSemester() {
        try_linkedlist.add(list.get(2));
        try_linkedlist.add(list.get(10));
        try_linkedlist.add(list.get(4));
        try_linkedlist.add(list.get(8));
        for (int i=0;i<try_linkedlist.size();++i)
            System.out.println(try_linkedlist.get(i));

        System.out.println("\nNext course of the same semester is "+try_linkedlist.get(0).getCourseName()
                +" is "+try_linkedlist.nextInSemester( index: 0).getData().getCourseName());
}
```

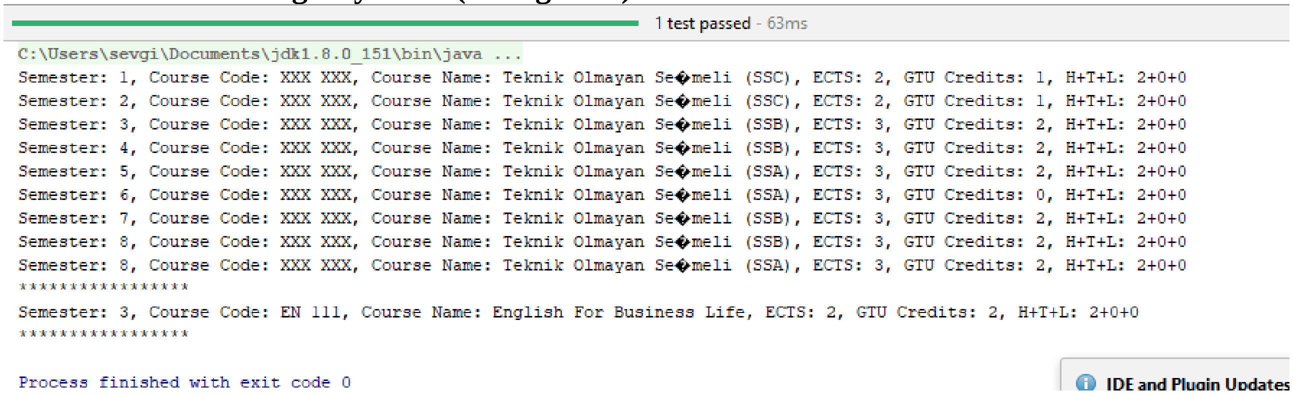
**Photo 3.1.10**

This test class has the list and try_linkedList variables.  The elements of variable the list is added to variable linkedList. After adding data of the try_linkedList is printed on the screen.After printing, the nextInSemester() method is called for index“0”.

## 3.2 Running Results

- **PART 1**

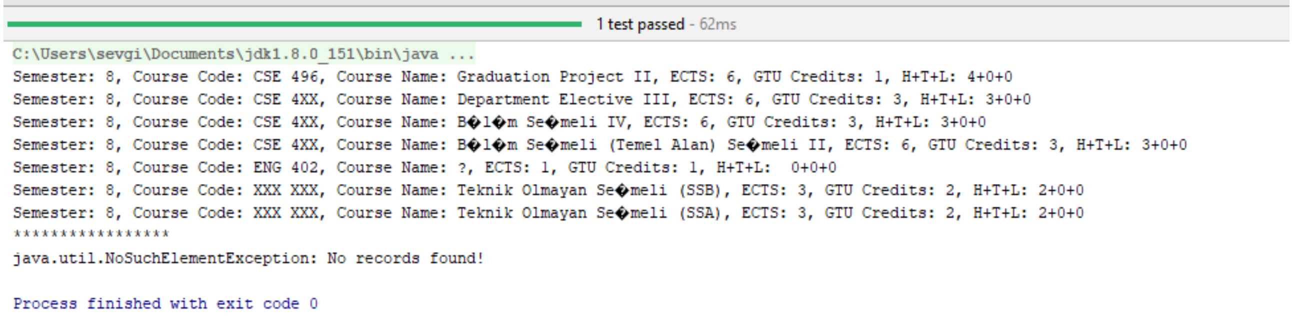
  1. **getByeCode(String code)**



```
1 test passed - 63ms
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\java ...
Semester: 1, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSC), ECTS: 2, GTU Credits: 1, H+T+L: 2+0+0
Semester: 2, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSC), ECTS: 2, GTU Credits: 1, H+T+L: 2+0+0
Semester: 3, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSB), ECTS: 3, GTU Credits: 2, H+T+L: 2+0+0
Semester: 4, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSB), ECTS: 3, GTU Credits: 2, H+T+L: 2+0+0
Semester: 5, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSA), ECTS: 3, GTU Credits: 2, H+T+L: 2+0+0
Semester: 6, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSA), ECTS: 3, GTU Credits: 0, H+T+L: 2+0+0
Semester: 7, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSB), ECTS: 3, GTU Credits: 2, H+T+L: 2+0+0
Semester: 8, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSB), ECTS: 3, GTU Credits: 2, H+T+L: 2+0+0
Semester: 8, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSA), ECTS: 3, GTU Credits: 2, H+T+L: 2+0+0
****************
Semester: 3, Course Code: EN 111, Course Name: English For Business Life, ECTS: 2, GTU Credits: 2, H+T+L: 2+0+0
****************

Process finished with exit code 0
```

The results can be compared to the inputs which are shown the Photo 3.1.1

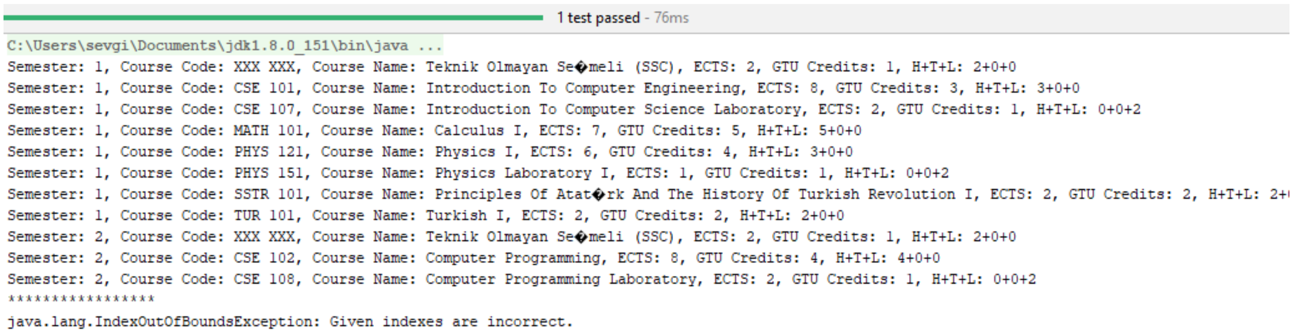
  2. **listSemesterCourses (int semester)**



```
1 test passed - 62ms
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\java ...
Semester: 8, Course Code: CSE 496, Course Name: Graduation Project II, ECTS: 6, GTU Credits: 1, H+T+L: 4+0+0
Semester: 8, Course Code: CSE 4XX, Course Name: Department Elective III, ECTS: 6, GTU Credits: 3, H+T+L: 3+0+0
Semester: 8, Course Code: CSE 4XX, Course Name: B�l�m Se�meli IV, ECTS: 6, GTU Credits: 3, H+T+L: 3+0+0
Semester: 8, Course Code: CSE 4XX, Course Name: B�l�m Se�meli (Temel Alan) Se�meli II, ECTS: 6, GTU Credits: 3, H+T+L: 3+0+0
Semester: 8, Course Code: ENG 402, Course Name: ?, ECTS: 1, GTU Credits: 1, H+T+L:  0+0+0
Semester: 8, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSB), ECTS: 3, GTU Credits: 2, H+T+L: 2+0+0
Semester: 8, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSA), ECTS: 3, GTU Credits: 2, H+T+L: 2+0+0
****************
java.util.NoSuchElementException: No records found!

Process finished with exit code 0
```

The results can be compared to the inputs which are shown the Photo 3.1.2

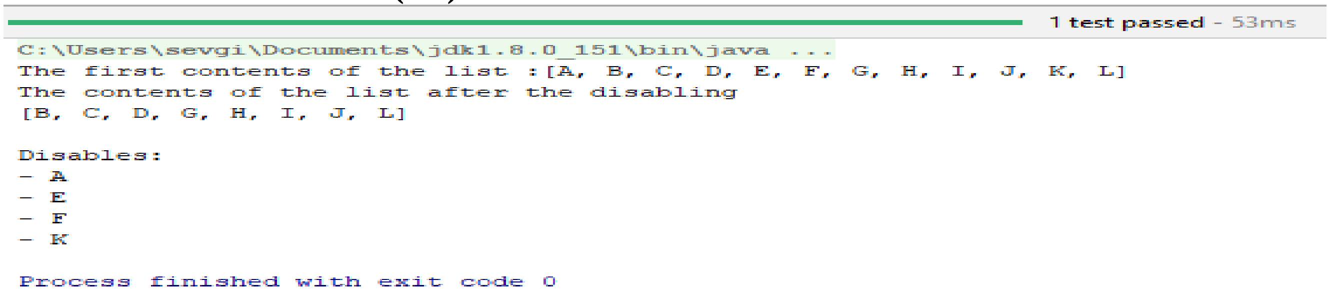
  3. **getByRange(int start_index, int last_index)**



```
1 test passed - 76ms
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\java ...
Semester: 1, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSC), ECTS: 2, GTU Credits: 1, H+T+L: 2+0+0
Semester: 1, Course Code: CSE 101, Course Name: Introduction To Computer Engineering, ECTS: 8, GTU Credits: 3, H+T+L: 3+0+0
Semester: 1, Course Code: CSE 107, Course Name: Introduction To Computer Science Laboratory, ECTS: 2, GTU Credits: 1, H+T+L: 0+0+2
Semester: 1, Course Code: MATH 101, Course Name: Calculus I, ECTS: 7, GTU Credits: 5, H+T+L: 5+0+0
Semester: 1, Course Code: PHYS 121, Course Name: Physics I, ECTS: 6, GTU Credits: 4, H+T+L: 3+0+0
Semester: 1, Course Code: PHYS 151, Course Name: Physics Laboratory I, ECTS: 1, GTU Credits: 1, H+T+L: 0+0+2
Semester: 1, Course Code: SSTR 101, Course Name: Principles Of Atat�rk And The History Of Turkish Revolution I, ECTS: 2, GTU Credits: 2, H+T+L: 2+
Semester: 1, Course Code: TUR 101, Course Name: Turkish I, ECTS: 2, GTU Credits: 2, H+T+L: 2+0+0
Semester: 2, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSC), ECTS: 2, GTU Credits: 1, H+T+L: 2+0+0
Semester: 2, Course Code: CSE 102, Course Name: Computer Programming, ECTS: 8, GTU Credits: 4, H+T+L: 4+0+0
Semester: 2, Course Code: CSE 108, Course Name: Computer Programming Laboratory, ECTS: 2, GTU Credits: 1, H+T+L: 0+0+2
****************
java.lang.IndexOutOfBoundsException: Given indexes are incorrect.
```

The results can be compared to the inputs which are shown the Photo 3.1.3

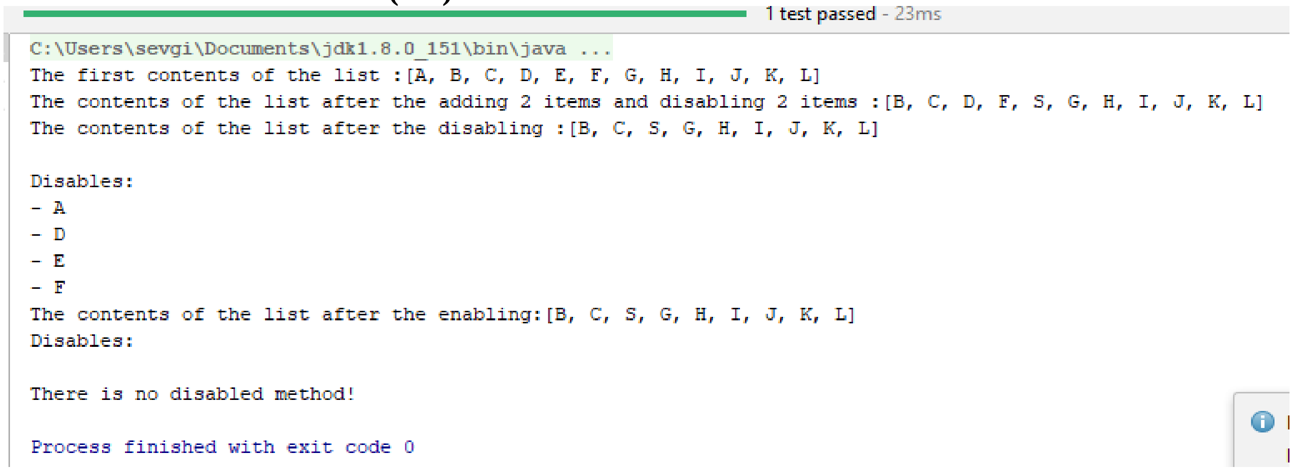
- **PART 2**
  1. **Disable(E e)**



```
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\java ...
The first contents of the list :[A, B, C, D, E, F, G, H, I, J, K, L]
The contents of the list after the disabling
[B, C, D, G, H, I, J, L]

Disables:
- A
- E
- F
- K

Process finished with exit code 0
```

The results can be compared to the inputs which are shown the Photo 3.1.4

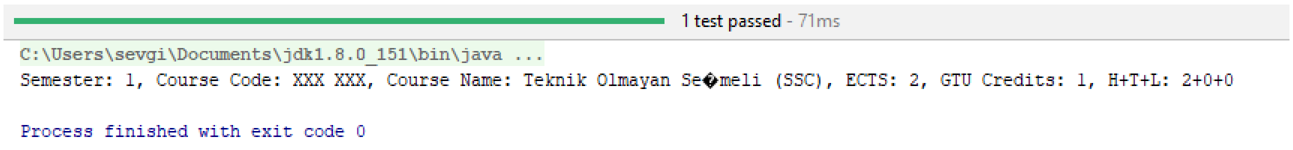
  2. **Enable(E e)**



```
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\java ...
The first contents of the list :[A, B, C, D, E, F, G, H, I, J, K, L]
The contents of the list after the adding 2 items and disabling 2 items :[B, C, D, F, S, G, H, I, J, K, L]
The contents of the list after the disabling :[B, C, S, G, H, I, J, K, L]

Disables:
- A
- D
- E
- F
The contents of the list after the enabling:[B, C, S, G, H, I, J, K, L]
Disables:

There is no disabled method!

Process finished with exit code 0
```

The results can be compared to the inputs which are shown the Photo 3.1.5

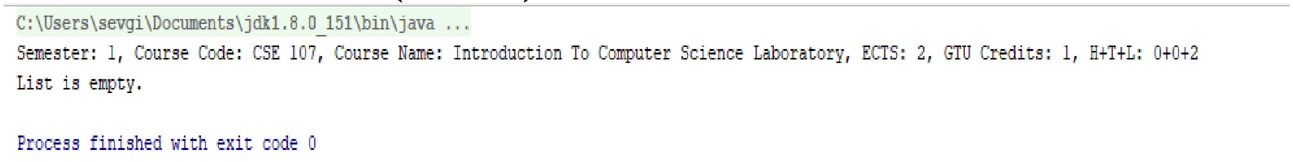
- **PART 3**
  1. **add(GTUCourse data)**



```
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\java ...
Semester: 1, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSC), ECTS: 2, GTU Credits: 1, H+T+L: 2+0+0

Process finished with exit code 0
```

The results can be compared to the inputs which are shown the Photo 3.1.6

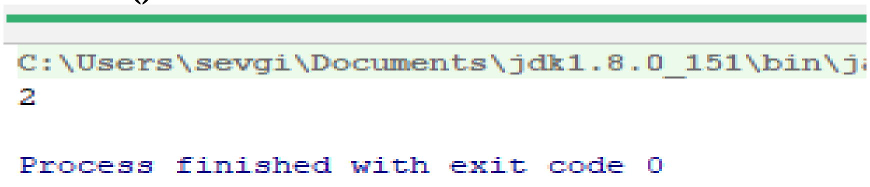
## 2. remove(int index)



```
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\java ...
Semester: 1, Course Code: CSE 107, Course Name: Introduction To Computer Science Laboratory, ECTS: 2, GTU Credits: 1, H+T+L: 0+0+2
List is empty.

Process finished with exit code 0
```

The results can be compared to the inputs which are shown the Photo 3.1.7

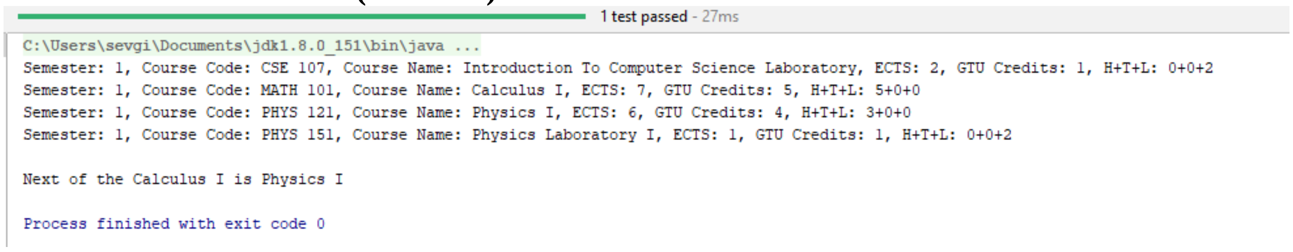
## 3. size()



```
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\j
2

Process finished with exit code 0
```

The results can be compared to the inputs which are shown the Photo 3.1.8

## 4. Next(int index)

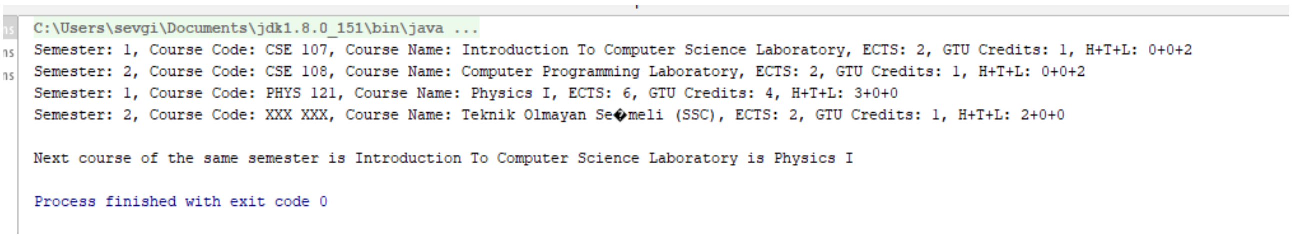


1 test passed - 27ms

```
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\java ...
Semester: 1, Course Code: CSE 107, Course Name: Introduction To Computer Science Laboratory, ECTS: 2, GTU Credits: 1, H+T+L: 0+0+2
Semester: 1, Course Code: MATH 101, Course Name: Calculus I, ECTS: 7, GTU Credits: 5, H+T+L: 5+0+0
Semester: 1, Course Code: PHYS 121, Course Name: Physics I, ECTS: 6, GTU Credits: 4, H+T+L: 3+0+0
Semester: 1, Course Code: PHYS 151, Course Name: Physics Laboratory I, ECTS: 1, GTU Credits: 1, H+T+L: 0+0+2

Next of the Calculus I is Physics I

Process finished with exit code 0
```

The results can be compared to the inputs which are shown the Photo 3.1.9

## 5. nextInSemester(int index)



```
C:\Users\sevgi\Documents\jdk1.8.0_151\bin\java ...
Semester: 1, Course Code: CSE 107, Course Name: Introduction To Computer Science Laboratory, ECTS: 2, GTU Credits: 1, H+T+L: 0+0+2
Semester: 2, Course Code: CSE 108, Course Name: Computer Programming Laboratory, ECTS: 2, GTU Credits: 1, H+T+L: 0+0+2
Semester: 1, Course Code: PHYS 121, Course Name: Physics I, ECTS: 6, GTU Credits: 4, H+T+L: 3+0+0
Semester: 2, Course Code: XXX XXX, Course Name: Teknik Olmayan Se�meli (SSC), ECTS: 2, GTU Credits: 1, H+T+L: 2+0+0

Next course of the same semester is Introduction To Computer Science Laboratory is Physics I

Process finished with exit code 0
```

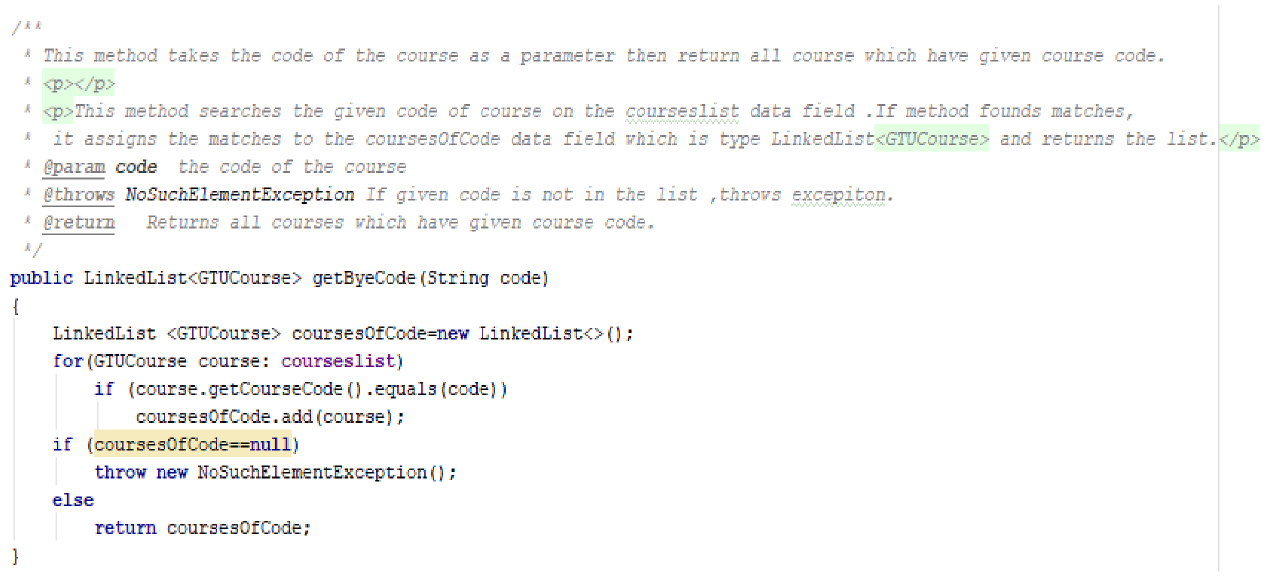
The results can be compared to the inputs which are shown the Photo 3.1.10

## 3.3 TIME COMPLEXITY ANALYSIS

All the time complexity is made according to the worst case.

- **Part 1**

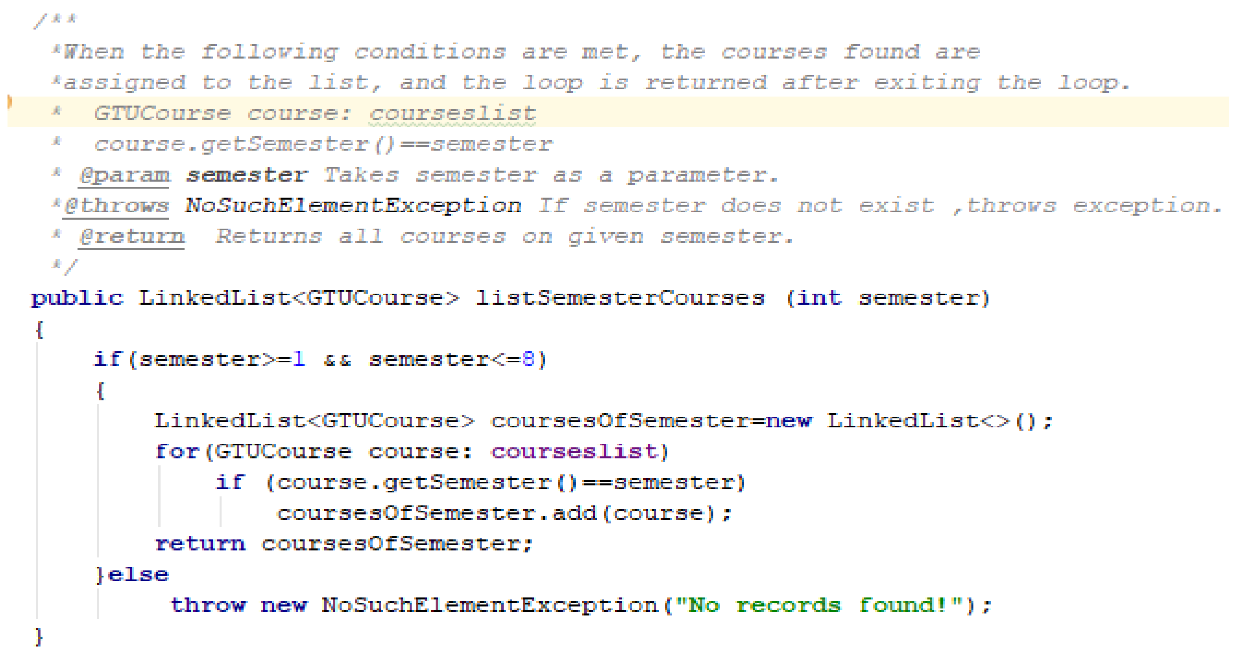
### 1. getByCode (string code)



```
/**
 * This method takes the code of the course as a parameter then return all course which have given course code.
 * <p></p>
 * <p>This method searches the given code of course on the courseslist data field .If method founds matches,
 *  it assigns the matches to the coursesOfCode data field which is type LinkedList<GTUCourse> and returns the list.</p>
 * @param code  the code of the course
 * @throws NoSuchElementException If given code is not in the list ,throws excepiton.
 * @return   Returns all courses which have given course code.
 */
public LinkedList<GTUCourse> getByeCode(String code)
{
    LinkedList <GTUCourse> coursesOfCode=new LinkedList<>();
    for(GTUCourse course: courseslist)
        if (course.getCourseCode().equals(code))
            coursesOfCode.add(course);
    if (coursesOfCode==null)
        throw new NoSuchElementException();
    else
        return coursesOfCode;
}
```

This function includes a for loop and this loop works coursesList.length time. So the time complexity of this function is **O(n).**

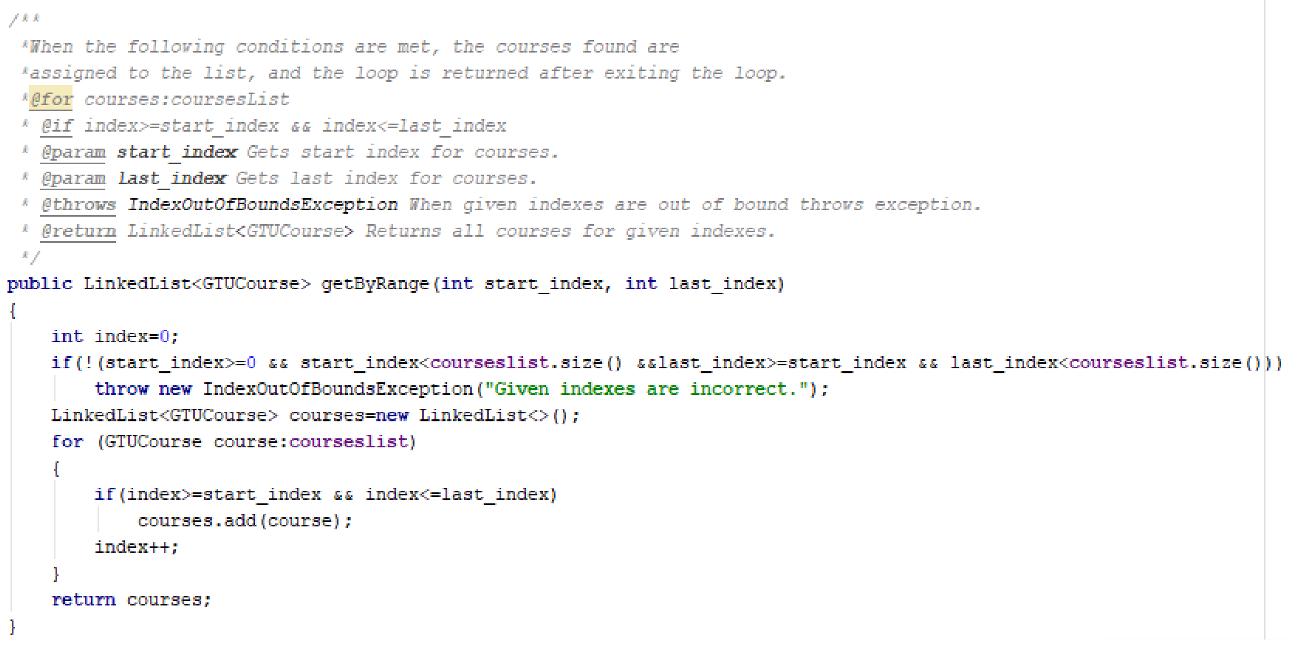
### 2. listSemesterCourses (int semester)



```
/**
 *When the following conditions are met, the courses found are
 *assigned to the list, and the loop is returned after exiting the loop.
 *  GTUCourse course: courseslist
 *  course.getSemester()==semester
 * @param semester Takes semester as a parameter.
 *@throws NoSuchElementException If semester does not exist ,throws exception.
 * @return  Returns all courses on given semester.
 */
public LinkedList<GTUCourse> listSemesterCourses (int semester)
{
    if(semester>=1 && semester<=8)
    {
        LinkedList<GTUCourse> coursesOfSemester=new LinkedList<>();
        for(GTUCourse course: courseslist)
            if (course.getSemester()==semester)
                coursesOfSemester.add(course);
        return coursesOfSemester;
    }else
        throw new NoSuchElementException("No records found!");
}
```

This function includes a for loop and this loop works coursesList.length time. So the time complexity of this function is **O(n).**

3. **getByRange(int start_index, int last_index)**

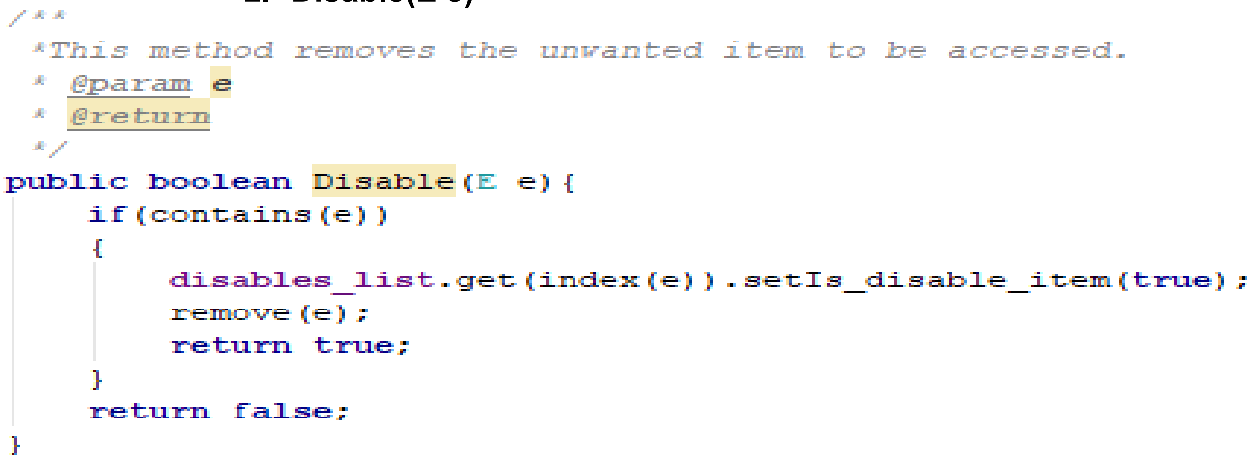


```java
/**
 *When the following conditions are met, the courses found are
 *assigned to the list, and the loop is returned after exiting the loop.
 *@for courses:coursesList
 * @if index>=start_index && index<=last_index
 * @param start_index Gets start index for courses.
 * @param last_index Gets last index for courses.
 * @throws IndexOutOfBoundsException When given indexes are out of bound throws exception.
 * @return LinkedList<GTUCourse> Returns all courses for given indexes.
 */
public LinkedList<GTUCourse> getByRange(int start_index, int last_index)
{
    int index=0;
    if(!(start_index>=0 && start_index<courseslist.size() &&last_index>=start_index && last_index<courseslist.size()))
        throw new IndexOutOfBoundsException("Given indexes are incorrect.");
    LinkedList<GTUCourse> courses=new LinkedList<>();
    for (GTUCourse course:courseslist)
    {
        if(index>=start_index && index<=last_index)
            courses.add(course);
        index++;
    }
    return courses;
}
```

This function includes a for loop and this loop works coursesList.length time. So the time complexity of this function is **O(n).**

- **PART 2**

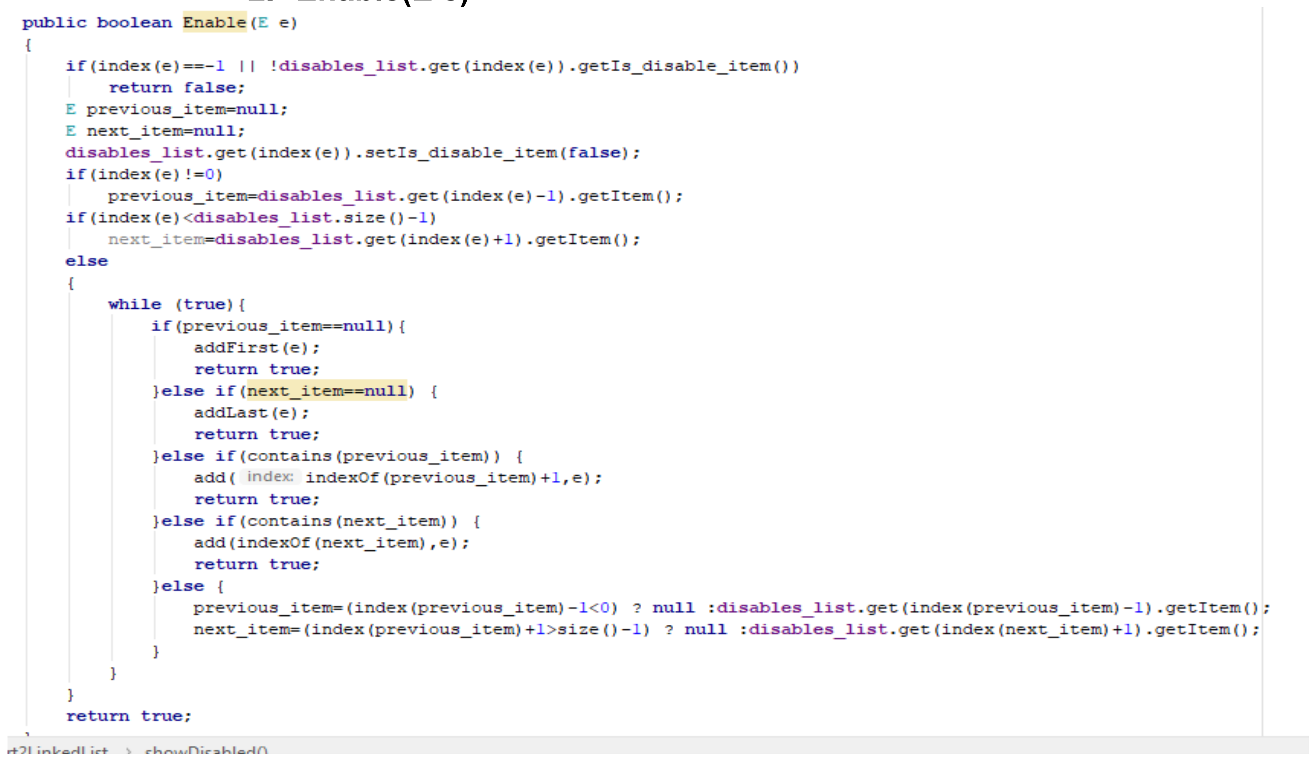
1. **Disable(E e)**



```java
/**
 *This method removes the unwanted item to be accessed.
 * @param e
 * @return
 */
public boolean Disable(E e){
    if(contains(e))
    {
        disables_list.get(index(e)).setIs_disable_item(true);
        remove(e);
        return true;
    }
    return false;
}
```

The time complexity of this function is **O(1).**

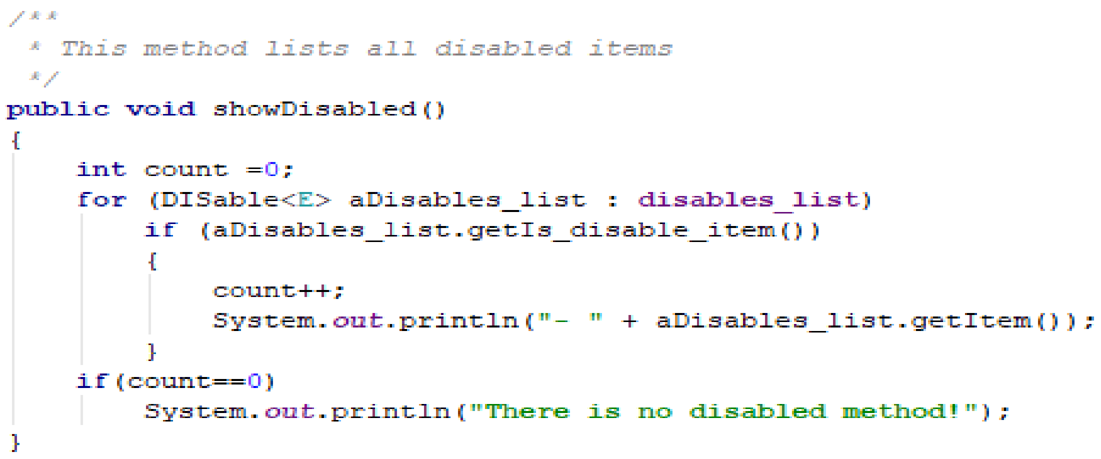
## 2. Enable(E e)



```
public boolean Enable(E e)
{
    if(index(e)==-1 || !disables_list.get(index(e)).getIs_disable_item())
        return false;
    E previous_item=null;
    E next_item=null;
    disables_list.get(index(e)).setIs_disable_item(false);
    if(index(e)!=0)
        previous_item=disables_list.get(index(e)-1).getItem();
    if(index(e)<disables_list.size()-1)
        next_item=disables_list.get(index(e)+1).getItem();
    else
    {
        while (true){
            if(previous_item==null){
                addFirst(e);
                return true;
            }else if(next_item==null) {
                addLast(e);
                return true;
            }else if(contains(previous_item)) {
                add( index: indexOf(previous_item)+1,e);
                return true;
            }else if(contains(next_item)) {
                add(indexOf(next_item),e);
                return true;
            }else {
                previous_item=(index(previous_item)-1<0) ? null :disables_list.get(index(previous_item)-1).getItem();
                next_item=(index(previous_item)+1>size()-1) ? null :disables_list.get(index(next_item)+1).getItem();
            }
        }
    }
    return true;
}
```

This function includes a loop and this loop works n time. So the time complexity of this function is **O(n).**

## 3. showDisabled()

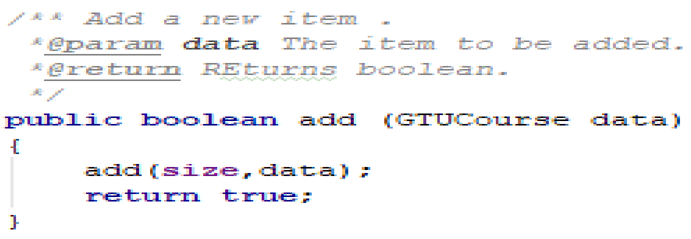


```
/**
 * This method lists all disabled items
 */
public void showDisabled()
{
    int count =0;
    for (DISable<E> aDisables_list : disables_list)
        if (aDisables_list.getIs_disable_item())
        {
            count++;
            System.out.println("- " + aDisables_list.getItem());
        }
    if(count==0)
        System.out.println("There is no disabled method!");
}
```

This function includes a loop and this loop works disable_list.length time. So the time complexity of this function is **O(n).**

- **PART 3**

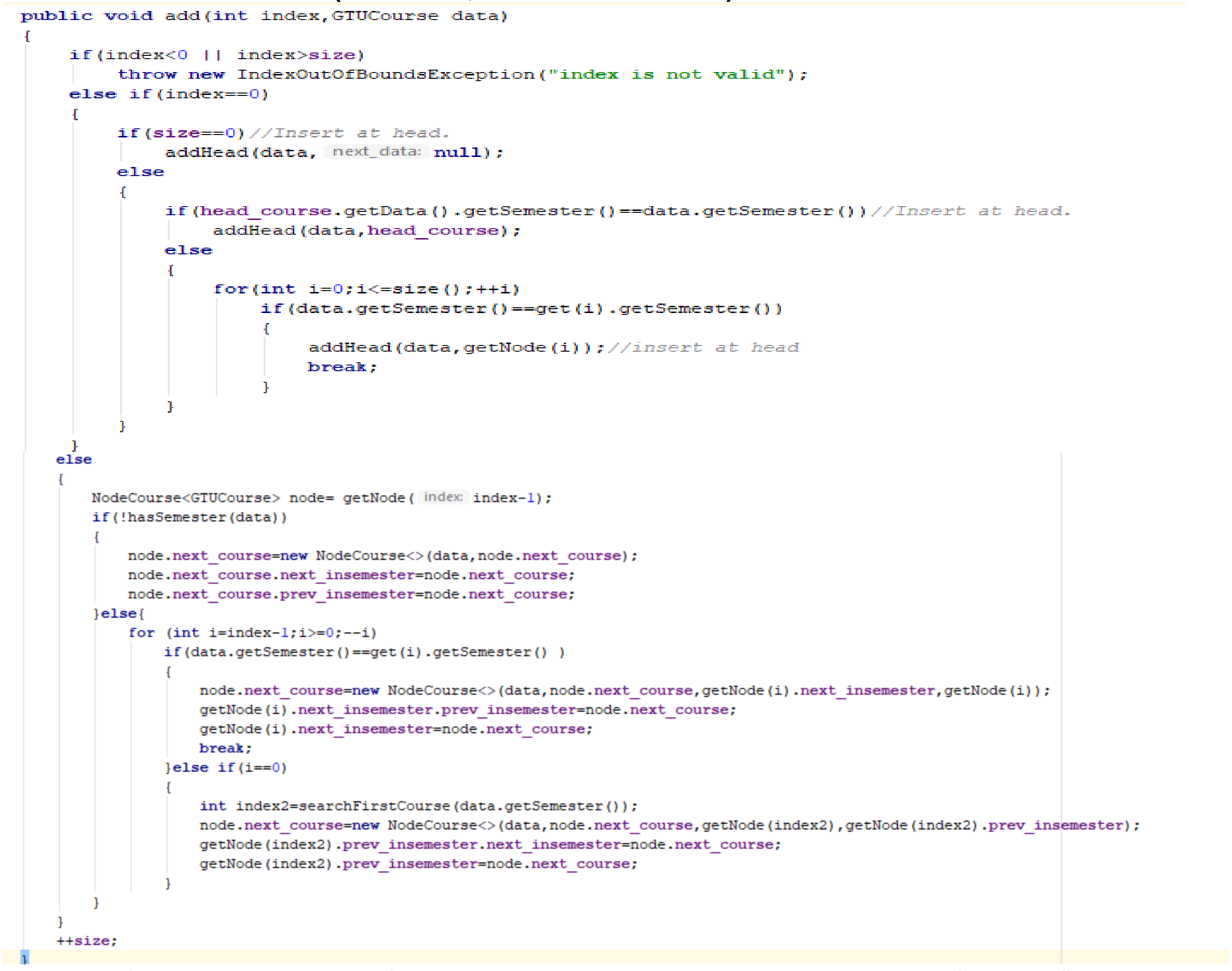
### 1. add( GTUCourse data)



```java
/** Add a new item .
 *@param data The item to be added.
 *@return REturns boolean.
 */
public boolean add (GTUCourse data)
{
    add(size,data);
    return true;
}
```

This function time complexity depends on other add method.Other add method time complexty is given below.

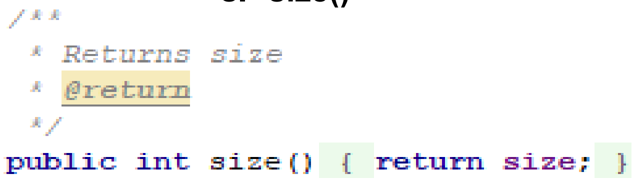
### 2. add(int index, GTUCourse data)



```java
public void add(int index,GTUCourse data)
{
    if(index<0 || index>size)
        throw new IndexOutOfBoundsException("index is not valid");
    else if(index==0)
    {
        if(size==0)//Insert at head.
            addHead(data, next_data: null);
        else
        {
            if(head_course.getData().getSemester()==data.getSemester())//Insert at head.
                addHead(data,head_course);
            else
            {
                for(int i=0;i<=size();++i)
                    if(data.getSemester()==get(i).getSemester())
                    {
                        addHead(data,getNode(i));//insert at head
                        break;
                    }
            }
        }
    }
    else
    {
        NodeCourse<GTUCourse> node= getNode( index: index-1);
        if(!hasSemester(data))
        {
            node.next_course=new NodeCourse<>(data,node.next_course);
            node.next_course.next_insemester=node.next_course;
            node.next_course.prev_insemester=node.next_course;
        }else{
            for (int i=index-1;i>=0;--i)
                if(data.getSemester()==get(i).getSemester() )
                {
                    node.next_course=new NodeCourse<>(data,node.next_course,getNode(i).next_insemester,getNode(i));
                    getNode(i).next_insemester.prev_insemester=node.next_course;
                    getNode(i).next_insemester=node.next_course;
                    break;
                }else if(i==0)
                {
                    int index2=searchFirstCourse(data.getSemester());
                    node.next_course=new NodeCourse<>(data,node.next_course,getNode(index2),getNode(index2).prev_insemester);
                    getNode(index2).prev_insemester.next_insemester=node.next_course;
                    getNode(index2).prev_insemester=node.next_course;
                }
        }
    }
    ++size;
}
```

This function includes a for loop and this loop includes searchFirstCourse () method(This method`s time complexity is **O(n)**).This loop works coursesList.length time. So the time complexity of this function is **O(n).O(n)==O(n^2).**

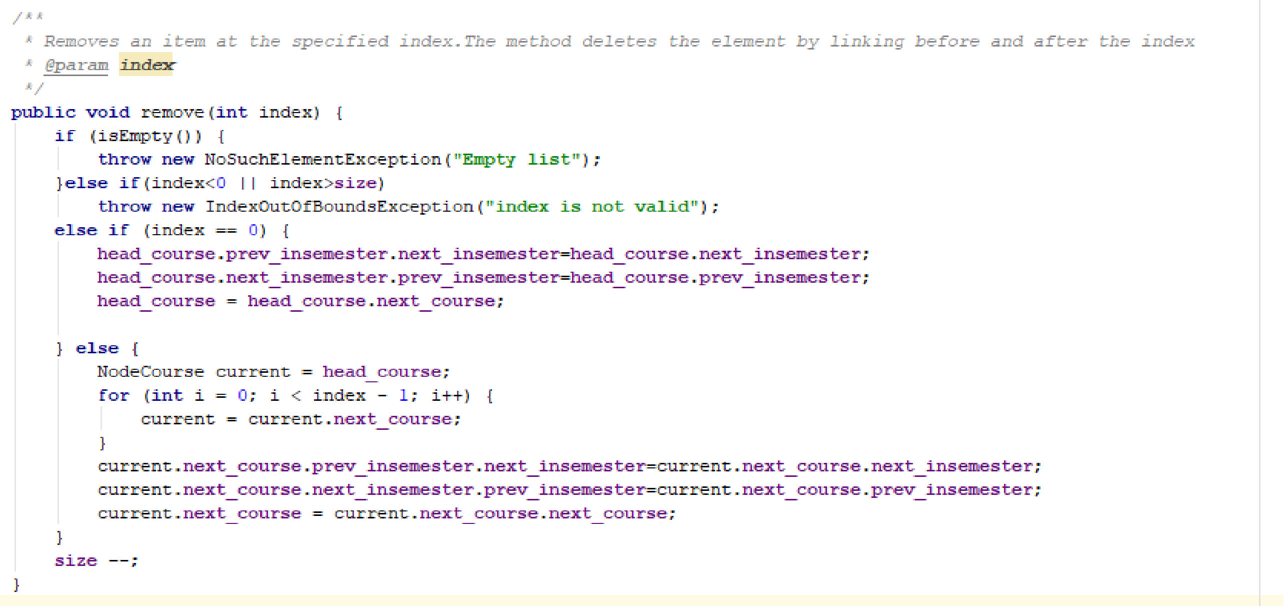
### 3. size()



```java
/**
 * Returns size
 * @return
 */
public int size() { return size; }
```

The time complexity of this function is **O(1).**

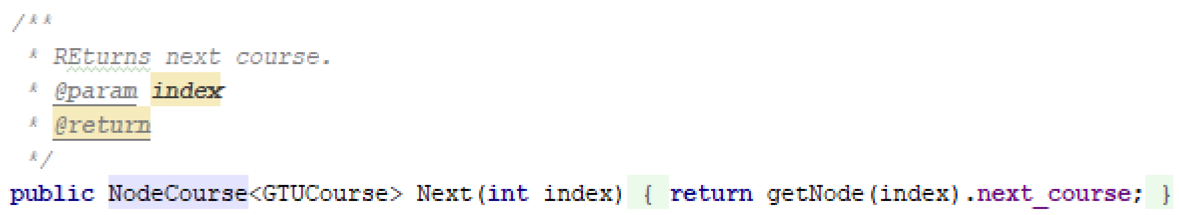
4. **remove(int index)**



```
/**
 * Removes an item at the specified index.The method deletes the element by linking before and after the index
 * @param index
 */
public void remove(int index) {
    if (isEmpty()) {
        throw new NoSuchElementException("Empty list");
    }else if(index<0 || index>size)
        throw new IndexOutOfBoundsException("index is not valid");
    else if (index == 0) {
        head_course.prev_insemester.next_insemester=head_course.next_insemester;
        head_course.next_insemester.prev_insemester=head_course.prev_insemester;
        head_course = head_course.next_course;

    } else {
        NodeCourse current = head_course;
        for (int i = 0; i < index - 1; i++) {
            current = current.next_course;
        }
        current.next_course.prev_insemester.next_insemester=current.next_course.next_insemester;
        current.next_course.next_insemester.prev_insemester=current.next_course.prev_insemester;
        current.next_course = current.next_course.next_course;
    }
    size --;
}
```

This function includes a loop and this loop works coursesList.length time. So the time complexity of this function is **O(n).**

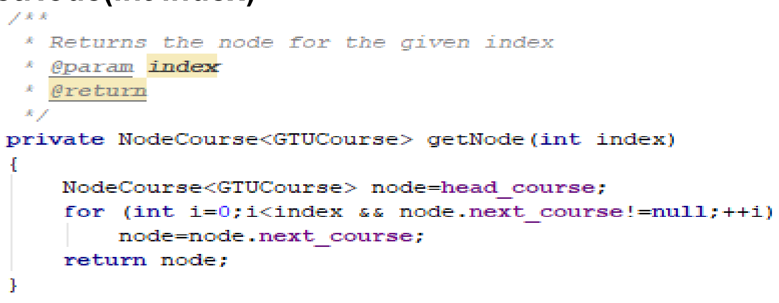
5. **Next(int index)**



```
/**
 * REturns next course.
 * @param index
 * @return
 */
public NodeCourse<GTUCourse> Next(int index) { return getNode(index).next_course; }
```

This function time complexity is depends the getNode().So the time complexity of this function is **O(n).**

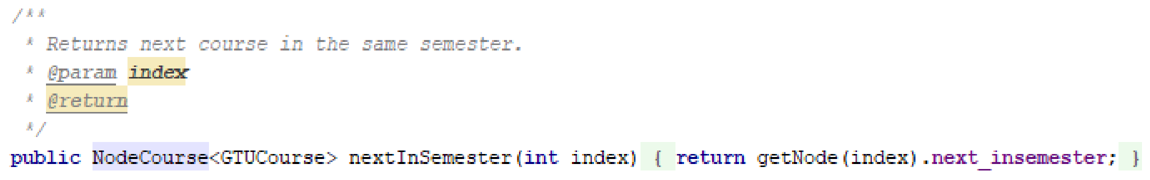
- **getNode(int index)**



```
/**
 * Returns the node for the given index
 * @param index
 * @return
 */
private NodeCourse<GTUCourse> getNode(int index)
{
    NodeCourse<GTUCourse> node=head_course;
    for (int i=0;i<index && node.next_course!=null;++i)
        node=node.next_course;
    return node;
}
```

This function includes a loop and this loop works index times. So the time complexity of this function is **O(n).**

6. **nextInSemester(int index)**



```
/**
 * Returns next course in the same semester.
 * @param index
 * @return
 */
public NodeCourse<GTUCourse> nextInSemester(int index) { return getNode(index).next_insemester; }
```

This function time complexity is depends the getNode().So the time complexity of this function is **O(n).**