

**Gebze Technical University
Computer Engineering**

CSE 222 - 2018 Spring

HOMEWORK 3 REPORT

**FERHAT ŞİRİN
161044080**

Course Assistant:

1 INTRODUCTION

1.1 Problem Definition

Part 1)

Controlling a list that has so many elements using a program. Easily access its data with the code and get a group of data.

Part 2)

Extra control for java LinkedList. User can easily block data without removing it after done it can be enabled.

Part 3)

Circular list inside a circular list. Elements that is in the same group is connected each other via circular list.

1.2 System Requirements

Part 1)

```
public Course getByCode(String code)
```

```
public LinkedList<Course> listSemesterCourses(int semester)
```

```
public LinkedList<Course> getByRange(int start, int end)
```

Course class is used with this function.

```
public Course(int semester,String code,String name,int ECTS,int GTU)
```

Course class constructor

Part 2)

```
public boolean enable(E obj)
```

```
public boolean disable(E obj)
```

```
public void showDisabled()
```

Part 3)

```
public Course add(Course obj)
```

```
public Course remove(Course obj)
```

```
public Course next()
```

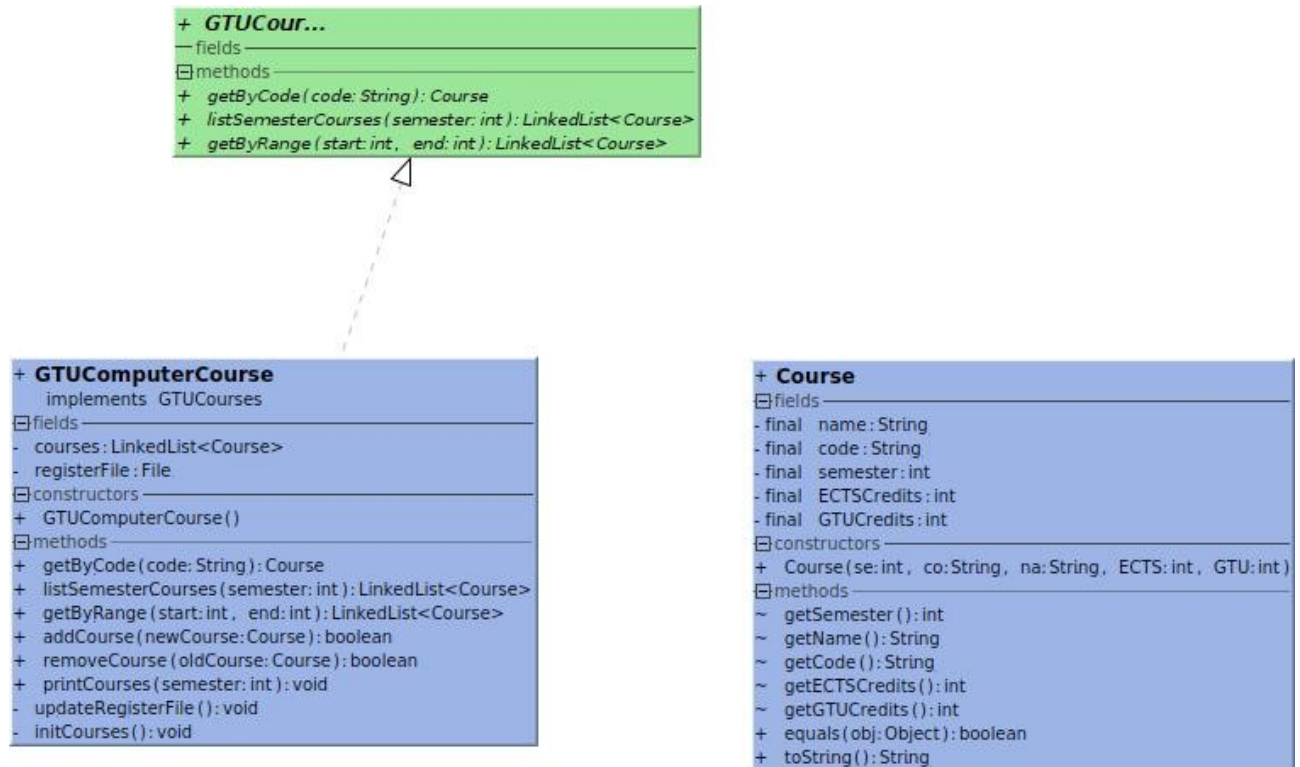
```
public Course nextInSemester()
```

Course Class is used with these functions like part 1.

2 METHOD

2.1 Class Diagrams

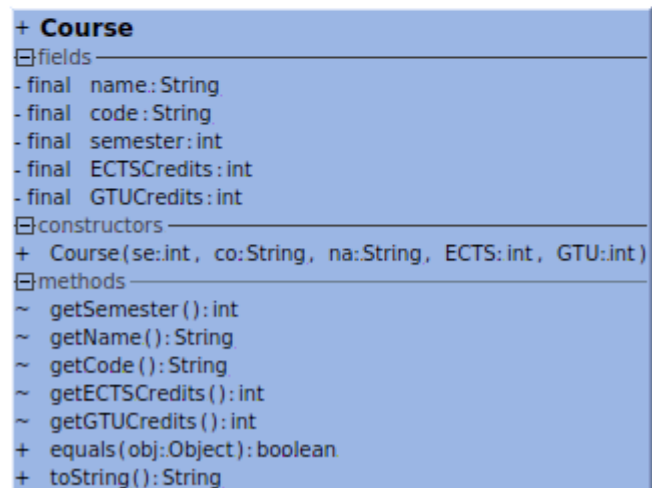
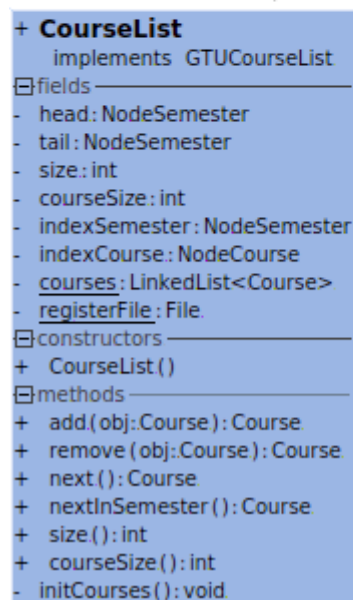
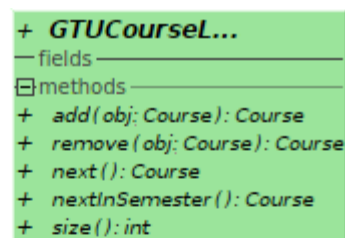
Part 1)



Part 2)

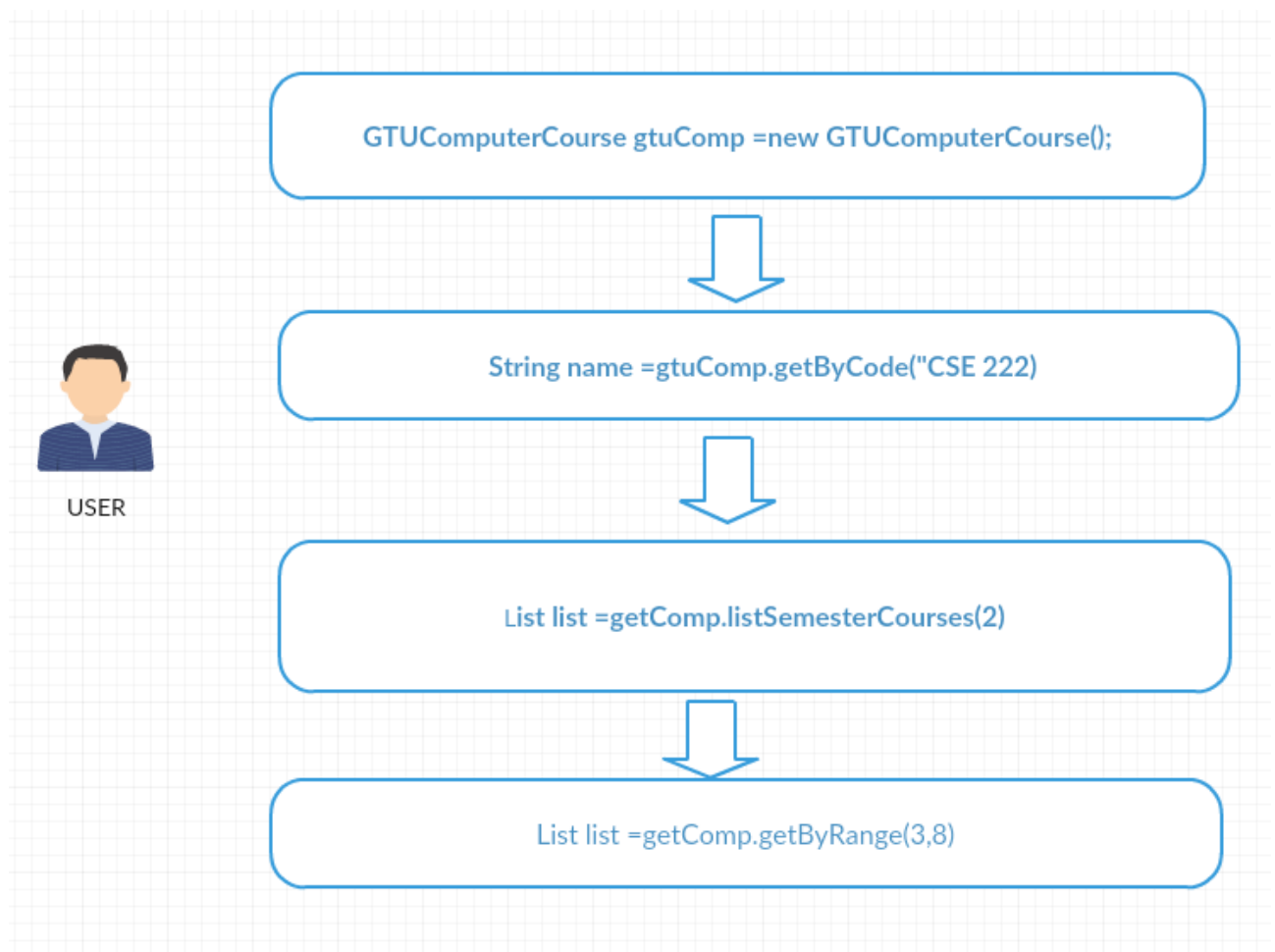


Part 3)

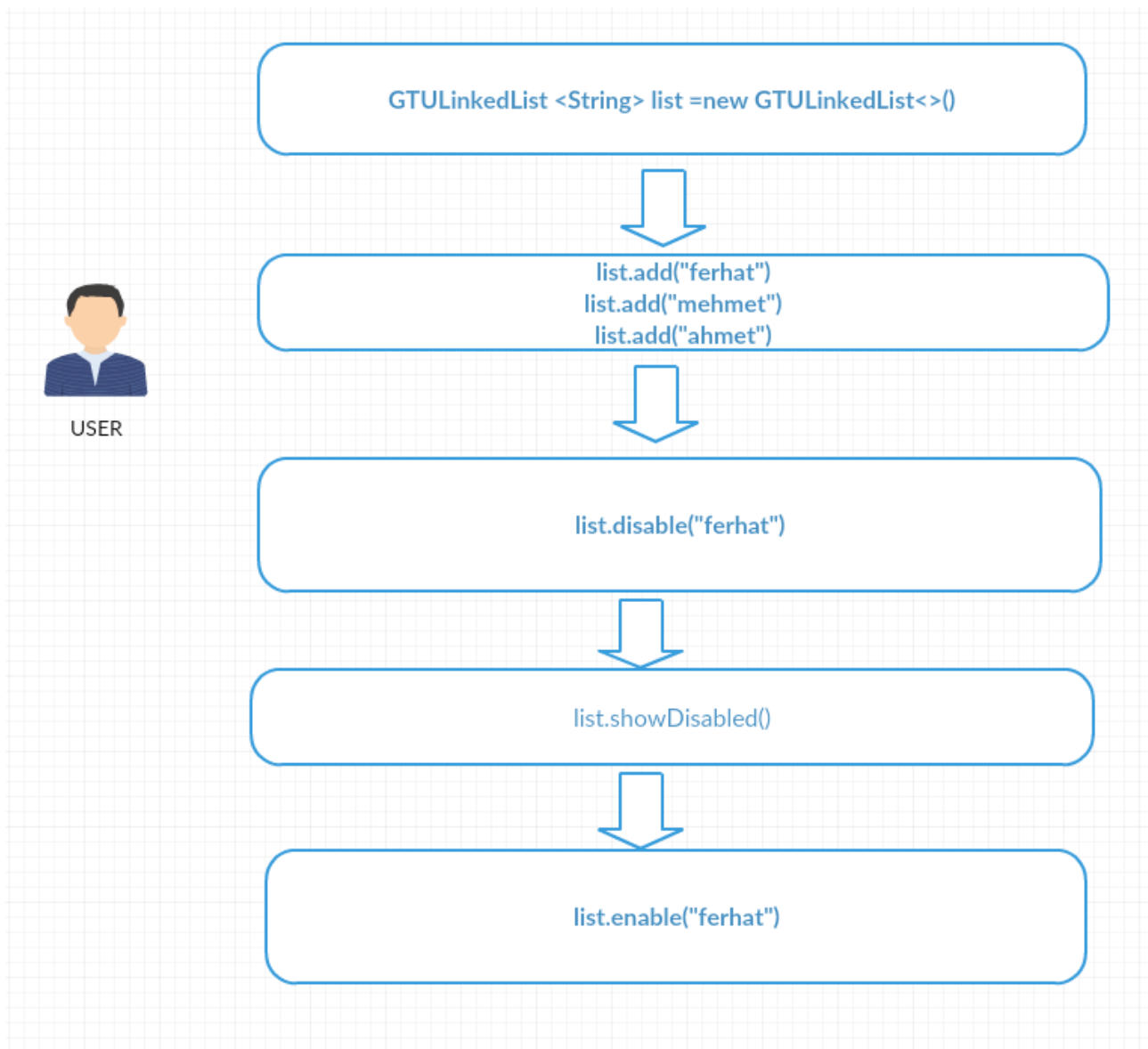


2.2 Use Case Diagrams

Part 1)



Part 2)



Part 3)



USER

```
CourseList list = new CourseList();
```

```
System.out.println("Printing 1. semester ");  
for (int i = 0; i < list.courseSize(); ++i)  
    System.out.println(list.next());
```

```
System.out.println("Printing 2. semester");  
list.nextInSemester();  
for (int i = 0; i < list.courseSize(); ++i)  
    System.out.println(list.next());
```

```
System.out.println("Removing CSE 102 Computer Programming  
course :");  
list.remove(new Course(2, "cse 102", "computer programming", 8, 4));
```

```
System.out.println("Adding CSE 102 Computer Programming  
course :");  
list.add(new Course(2, "cse 102", "computer programming", 8, 4))
```

```
System.out.println("Printing all list elements :");  
for (int i = 0; i < list.size(); ++i) {  
    for (int j = 0; j < list.courseSize(); ++j) {  
        System.out.println(list.next());  
    }  
    list.nextInSemester();  
}
```

2.3 Problem Solution Approach

Part 1)

Java LinkedList used in this part. Course class is written to make data transfer easy among the function. Course class includes everything about a course.

GTUComputerCourse reads data from file and help user to use information in that file via getByCode, listSemesterCourse and getByRange.

Part 2)

Java LinkedList is extended in this part. Super class features stay as default. Disable enable and showDisabled added as helper. Another LinkedList data structure is used to hold disabled list data.

Part 3)

A data structure like CircularLinkedList is created in this part. Data is held via Nodes. There are two nodes which is NodeSemester and NodeCourses which is inside NodeSemester to hold data. NodeSemester hold semester first data (head) and NodeCourses hold the corresponding courses data.

2.4 Asymptotic Notations

Part 1)

```
public Course getByCode(String code){
    for(int i=0;i<courses.size();++i){
        if(courses.get(i).getCode().equalsIgnoreCase(code)){
            return courses.get(i);
        }
    }
    throw new NoSuchElementException(code);
}
```

$O(n)$

```
public LinkedList<Course> listSemesterCourses(int semester){
    if(semester < 1 || 8 < semester){
        throw new ArrayIndexOutOfBoundsException(semester);
    }
    int i;
    for(i=0; courses.get(i).getSemester() != semester;++i){
    }
    LinkedList<Course> list =new LinkedList<>();
    for(;i < courses.size() && courses.get(i).getSemester() == semester ;
    ++i)
        list.add(courses.get(i));
    return list;
}
```


n number of all course and k is number of that semester course

$\max(\theta(n) + \theta(k))$

```
public LinkedList<Course> getByRange(int start, int end){
    if(start < 0 || end <= start || courses.size() < end){
        throw new ArrayIndexOutOfBoundsException(start);
    }
    LinkedList<Course> list =new LinkedList<>();
    for(int i =start; i<=end ;++i)
        list.add(courses.get(i));
    return list;
}
```

n =end – start

$\theta(n)$

Part 2)

```
public boolean disable(E obj){
    int i =indexOf(obj);
    if(i != -1){
        remove(i);
        disabledList.add(new Node(obj,i));
        return true;
    }
    return false;
}
```

indexOf(obj) is $O(n)$

disable itself $\theta(1)$

result is $O(n)$

```
public boolean enable(E obj){
    for(int i=0;i<disabledList.size();++i) {
        if (disabledList.get(i).data.equals(obj)){
            add(disabledList.get(i).index,obj);
            disabledList.remove(i);
            return true;
        }
    }
    return false;
}
```

disabledList size n

$O(n)$

```
public void showDisabled(){
    for(int i=0;i<disabledList.size();++i){
        System.out.println(disabledList.get(i).data);
    }
}
```

$\theta(n)$

Part 3)

```
public Course add(Course obj) {
    if(head ==null){
        head =new NodeSemester(obj);
        tail =head;
        indexSemester =head;
        indexCourse =indexSemester.courseHead;
        courseSize =indexSemester.courseSize;
        ++size;
        return obj;
    }
    NodeSemester semester =head;
    for(int i=0;i< size;++i){
        if(semester.courseHead.data.getSemester() == obj.getSemester()) {
            NodeSemester.NodeCourse newCourse =new
NodeSemester.NodeCourse(obj);
            semester.courseTail.next =newCourse;
            newCourse.next =semester.courseHead;
            semester.courseTail =newCourse;
            ++semester.courseSize;
            return obj;
        }
        semester =semester.nextSemester;
    }
    semester =new NodeSemester(obj);
    tail.nextSemester =semester;
    semester.nextSemester =head;
    tail =semester;
    ++size;
    return obj;
}
```

list size =n and different semester added

Tworst = $O(n)$

list size =0

Tbest = $\theta(1)$

```
public Course remove(Course obj) throws NoSuchElementException {
    if(head != null){
        NodeSemester temp =head;
        for(int i=0;i<size;++i){
            if(temp.courseHead.data.getSemester() == obj.getSemester()){
                if(temp.courseHead.data.equals(obj)){
                    if(indexCourse == temp.courseHead)
                        indexCourse =indexCourse.next;
                    temp.courseHead =temp.courseHead.next;
                    temp.courseTail.next =temp.courseHead;
                    --temp.courseSize;
                    return obj;
                }
                NodeSemester.NodeCourse course =temp.courseHead;
            }
        }
    }
}
```

```

        for(int j=0; j< temp.courseSize-1;++j) {
            if(course.next.data.equals(obj)){
                if(course.next == temp.courseTail){
                    temp.courseTail =course;
                }
                course.next =course.next.next;
                --temp.courseSize;
                return obj;
            }
            course =course.next;
        }
    }
    temp =temp.nextSemester;
}
}
throw new NoSuchElementException(obj.toString());
}

```

Semester size n course size k

$T_{worst} = O(n*k)$

$T_{best} = \theta(1)$

```

public Course next() {
    NodeSemester.NodeCourse old =indexCourse;
    indexCourse =indexCourse.next;
    return old.data;
}

```

$\theta(1)$

```

public Course nextInSemester() {
    NodeSemester old =indexSemester;
    indexSemester =indexSemester.nextSemester;
    indexCourse =indexSemester.courseHead;
    courseSize =indexSemester.courseSize;
    return old.courseHead.data;
}

```

$\theta(1)$

3 RESULT

3.1 Test Cases

Part 1)

All test cases in GTUCourseTest main function

Part 2)

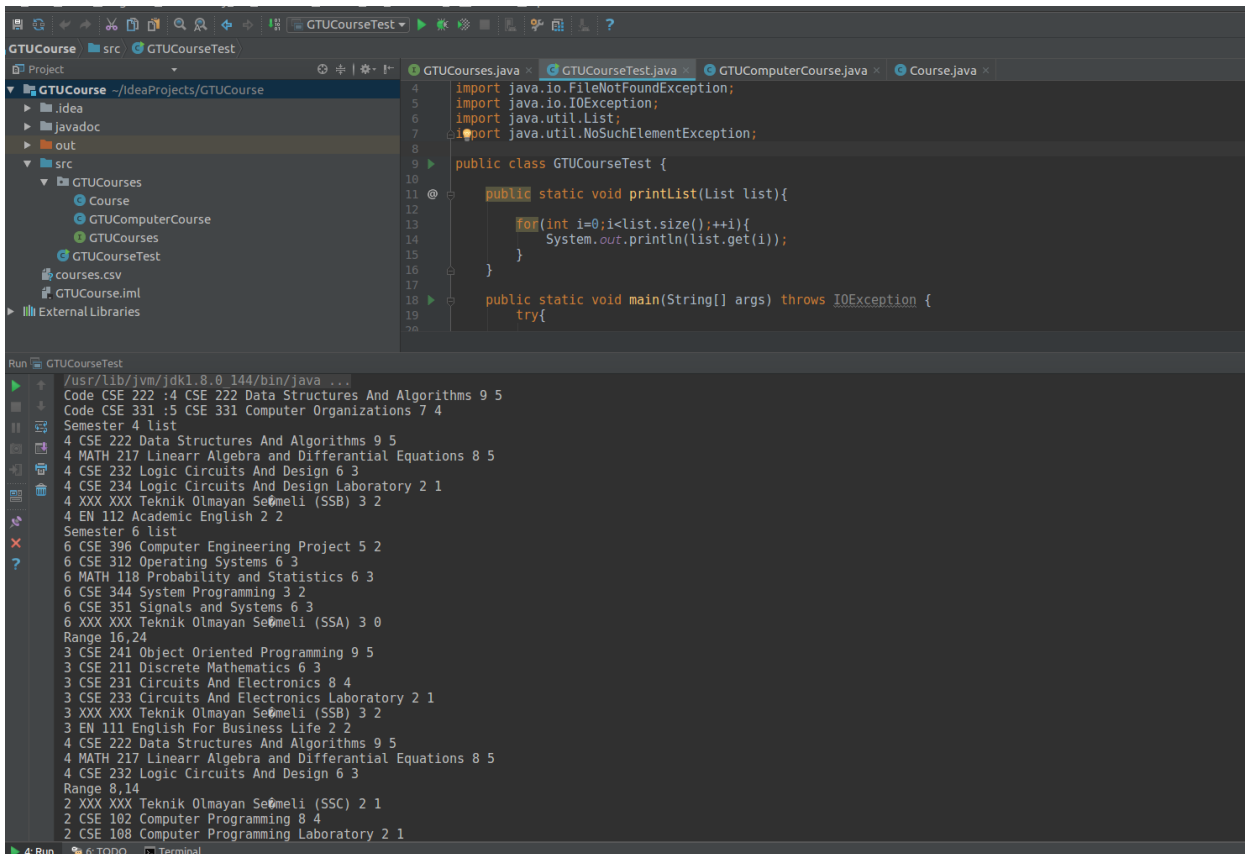
All test cases in GtuLinkedListTest main function

Part 3)

All test cases in CourseListTest main function

3.2 Running Results

Part 1)



The screenshot displays an IDE with a project named 'GTUCourse'. The left sidebar shows the project structure, including 'src' and 'External Libraries'. The main editor shows the 'GTUCourseTest.java' file with the following code:

```
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6 import java.util.List;
7 import java.util.NoSuchElementException;
8
9 public class GTUCourseTest {
10
11     public static void printList(List list){
12         for(int i=0;i<list.size();++i){
13             System.out.println(list.get(i));
14         }
15     }
16
17     public static void main(String[] args) throws IOException {
18         try{
19
20
```

The bottom panel shows the 'Run' output, which lists various course details and their associated marks:

```
Run: GTUCourseTest
/usr/lib/jvm/jdk1.8.0_144/bin/java ...
Code CSE 222 :4 CSE 222 Data Structures And Algorithms 9 5
Code CSE 331 :5 CSE 331 Computer Organizations 7 4
Semester 4 list
4 CSE 222 Data Structures And Algorithms 9 5
4 MATH 217 Linearr Algebra and Differantial Equations 8 5
4 CSE 232 Logic Circuits And Design 6 3
4 CSE 234 Logic Circuits And Design Laboratory 2 1
4 XXX XXX Teknik Olmayan Se0meli (SSB) 3 2
4 EN 112 Academic English 2 2
Semester 6 list
6 CSE 396 Computer Engineering Project 5 2
6 CSE 312 Operating Systems 6 3
6 MATH 118 Probability and Statistics 6 3
6 CSE 344 System Programming 3 2
6 CSE 351 Signals and Systems 6 3
6 XXX XXX Teknik Olmayan Se0meli (SSA) 3 0
Range 16,24
3 CSE 241 Object Oriented Programming 9 5
3 CSE 211 Discrete Mathematics 6 3
3 CSE 231 Circuits And Electronics 8 4
3 CSE 233 Circuits And Electronics Laboratory 2 1
3 XXX XXX Teknik Olmayan Se0meli (SSB) 3 2
3 EN 111 English For Business Life 2 2
4 CSE 222 Data Structures And Algorithms 9 5
4 MATH 217 Linearr Algebra and Differantial Equations 8 5
4 CSE 232 Logic Circuits And Design 6 3
Range 8,14
2 XXX XXX Teknik Olmayan Se0meli (SSC) 2 1
2 CSE 102 Computer Programming 8 4
2 CSE 108 Computer Programming Laboratory 2 1
```

Part 2)

```
GTULinkedList.java
1 import GTULinkedList.GTULinkedList;
2
3 import java.util.List;
4
5 public class GTULinkedListTest {
6
7     @ public static void printList(List list){
8         for(int i=0;i<list.size();++i){
9             System.out.println(list.get(i).toString());
10        }
11    }
12
13    public static void main(String[] args){
14
15        GTULinkedList<String> list =new GTULinkedList<>();
16
17        list.add("ferhat");
18        list.add("mehmet");
19        list.add("ahmet");
20
21        System.out.println("Printing list :");
22        printList(list);
23    }
24
25    GTULinkedListTest -> main()

```

```
run GTULinkedListTest
+ /usr/lib/jvm/jdk1.8.0_144/bin/java ...
+ Printing list :
+ ferhat
+ mehmet
+ ahmet
+ After disabling 'ferhat' show disabled:
+ ferhat
+ printList :
+ mehmet
+ ahmet
+ After enabling printList:
+ ferhat
+ mehmet
+ ahmet
+ After disabling 'ahmet' show disabled:
+ ahmet
+ printList :
+ ferhat
+ mehmet
+ After enabling printList:
+ ferhat
+ mehmet
+ ahmet
+ Process finished with exit code 0

```

Part 3)

```
CourseList.java
7 public class CourseListTest {
8
9     public static void main(String[] args){
10
11         try {
12             CourseList list = new CourseList();
13
14             System.out.println("Printing 1. semester ");
15             for (int i = 0; i < list.courseSize(); ++i)
16                 System.out.println(list.next());
17             System.out.println("After list.nextInSemester() printing 2. semester");
18             list.nextInSemester();
19             for (int i = 0; i < list.courseSize(); ++i)
20                 System.out.println(list.next());
21
22             System.out.println("Removing CSE 102 Computer Programming course :");
23             list.remove(new Course( sem:2, cos:"cse 102", na:"computer programming", ECTS: 8, GTU: 4));
24             System.out.println("Semester 2 :");
25             for (int i = 0; i < list.courseSize(); ++i)
26                 System.out.println(list.next());
27         }
28     }
29
30     CourseListTest -> main()

```

```
run CourseListTest
+ /usr/lib/jvm/jdk1.8.0_144/bin/java ...
+ Printing 1. semester
+ 1 XXX XXX Teknik Olmayan Seemeli (SSC) 2 1
+ 1 CSE 101 Introduction To Computer Engineering 8 3
+ 1 CSE 107 Introduction To Computer Science Laboratory 2 1
+ 1 MATH 101 Calculus I 7 5
+ 1 PHYS 121 Physics I 6 4
+ 1 PHYS 151 Physics Laboratory I 1 1
+ 1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2
+ 1 TUR 101 Turkish I 2 2
+ After list.nextInSemester() printing 2. semester
+ 2 XXX XXX Teknik Olmayan Seemeli (SSC) 2 1
+ 2 CSE 102 Computer Programming 8 4
+ 2 CSE 108 Computer Programming Laboratory 2 1
+ 2 MATH 102 Calculus II 7 5
+ 2 PHYS 122 Physics II 6 4
+ 2 PHYS 152 Physics Laboratory II 1 1
+ 2 SSTR 102 Principles Of Atatürk And The History Of Turkish Revolution II 2 2
+ 2 TUR 102 Turkish II 2 2
+ Removing CSE 102 Computer Programming course :
+ Semester 2 :
+ 2 XXX XXX Teknik Olmayan Seemeli (SSC) 2 1
+ 2 CSE 108 Computer Programming Laboratory 2 1
+ 2 MATH 102 Calculus II 7 5
+ 2 PHYS 122 Physics II 6 4
+ 2 PHYS 152 Physics Laboratory II 1 1
+ 2 SSTR 102 Principles Of Atatürk And The History Of Turkish Revolution II 2 2

```