**Gebze Technical University**

**Computer Engineering**

**CSE 222 - 2018 Spring**

**HOMEWORK X REPORT**

**STUDENT NAME**

**STUDENT NUMBER**

Course Assistant:

# INTRODUCTION

## Problem Definition

PART 1. Bilgisayar mühendisliği lisans derslerinin yer alan bir dosyadaki bütün derslerin ve derslere ait özelliklerinin hepsini içeren bir Linked List oluşturulmalı. Liste bütün dersleri tutmalı ve derslerin dosyada yer alan bütün özellikleri de yer almalıdır. Ve bu listedeki derslerle alakalı şu işlemler yapılmalıdır:

1. getByCode metodu: Bir ders verilen kodu ile bulunmalıdır.
2. listSemesterCourses: Verilen semestırdaki bütün dersleri göstermeli.
3. getByRange(int start\_index, int last\_index): Kurslar ayrıca başlangıç ve bitiş index numalarına göre bulunabilmelidir.

PART 2: disable(), enable() and showDisabled() metotları ile Part 1’de oluşturulan listeden dersler gizlenmeli ve sonra tekrar geri getirilmeli hatta gizlenmiş olan dersler get metoduyla gösterilmemelidir.

PART 3: Yeni bir list oluşturularak Listedeki dersler aynı semesterda olanlar birbirini göstermek şartıyla dairesel olarak listelenmelidir.

## System Requirements

Give detailed system requirements

# METHOD

## Class Diagrams

Add class diagrams.

## Use Case Diagrams

Add use case diagrams if required.

## Other Diagrams (optional)

Add other diagrams if required.

## Problem Solution Approach:

Ödev Part 1 Part 2 ve Part 3 olmak üzere 3 bölümden ve 3 ayrı mainden oluşmaktadır. Part 1’de courseList adında bir LinkListe kursların yer aldığı csv file okundu. Ardından bütün dersler şu metotlarla gösterildi:

courselist.printAllCourses();  
 System.*out*.println(courselist.size());

courselist.remove("YXX XXX");

Yukardıdaki kod ile kurs kodu “yxx xxx” olan ders listeden silindi. Ayrıca if döngüsü sayesinde belirli bir koda ait bir kurs listeden kolaylıkla bulunabildi. Belirli bir semestırdaki kurslar da listeden çıkarılıp ekrana yazdırıldı.

GTUComputerEngineeringLinkList semesterlinklist = courselist.listSemesterCourses(2);  
semesterlinklist.printAllCourses();

Kısacası getByCode, listSemesterCoursesall ve getByRange(int start\_index, int last\_index) metotları sayesinde kurslar kodlarına göre aranıp bulundu, silindi ya da herhangi bir semestıra ait kurslar bulunabildi. Ayrıca başlangıç ve bitiş indexlerine göre de kurslar bulunabildi.

Part 2’de ise;

MyCustomLinkedList sınıfı oluşturularak burada disabledItems adında, disabled olanların listesini tutması için bir liste oluşturuldu.

disabledItems.add(index);

ile listeye itemsleri eklenir.

disabledItems.remove(disabledItems.indexOf(index));

ile itemsler listeden silinir.

Disabled olan item eğer istenirse hata fırlatılır:

@Override  
public E set(int index, E element) throws ArrayIndexOutOfBoundsException  
{  
 if( disabledItems.contains(index) )  
 {  
 throw new ArrayIndexOutOfBoundsException("Index is disabled"); // E�er set istenilen index disabled ise hata f�rlat�r.  
 }  
 return (E)(super.set(index, element));  
}

Disabled olan itemler list.size()’dan da çıkarılarak yeni listenin genişliği bulundu.

public int size() {  
 // *TODO Auto-generated method stub* return super.size()-disabledItems.size(); //Linkedlist eleman say�s�ndan disabled eleman say�s�n� ��kard�m.  
}

ve remove ile disabled olan itemsler tekrar enable olabildi:

disabledItems.remove(disabledItems.indexOf(index)); //index'i disabled listesinden sildim.

PART 3:

Zorlanıldı.. Tam olarak sonuç elde edilemedi. Kodlamalarda bir sorun yaşandı. Öncelikle nodların aynı semesterda olanları göstermesi gerekiyordu. File load UTF-8 hatası aldım. Sorunu çözedim. BU soruda sisteme yine tüm kurslar dosyadan yüklendi ve

Iterator<CourseNode> it = myCircularCustomList.iterator();

İterator tanımlandı. MultiCirkularLinkedList SemesterIterator ve Iteratörü implement etti.

public class MyMultiCircularLinkedList implements Iterator<CourseNode>, SemesterIterator<CourseNode>{

private CourseNode head;  
private int nextIndex = 0;  
private int sizeCount = 0;

CourseNode, nextIndex e SizeCount adında değişkenler tanımlandı. Hashtable ile semestırların cirkular linked list referansları tutuldu.

private Hashtable<Integer,ArrayList<CourseNode>> circularLinkedListRefs = new Hashtable<Integer,ArrayList<CourseNode>>()

ve tek tek nodalar eklendi.

if(!circularLinkedListRefs.containsKey(node.getSemester()))  
 circularLinkedListRefs.put(node.getSemester(), new ArrayList<>());  
circularLinkedListRefs.get(node.getSemester()).add(node);  
  
sizeCount++;

Bu döngüyle aynı semestırda olan derslerden arraylist oluşturuldu.

Bu döngüde ise custom iteratör döndürüldü:

public SemesterIterator<CourseNode> nextInSemesterIterator(int semester){ //Custom iterator döndürüldü.  
 selectedNextSemester = semester;  
 currentSemesterIndex = 0;  
 return this;   
 }

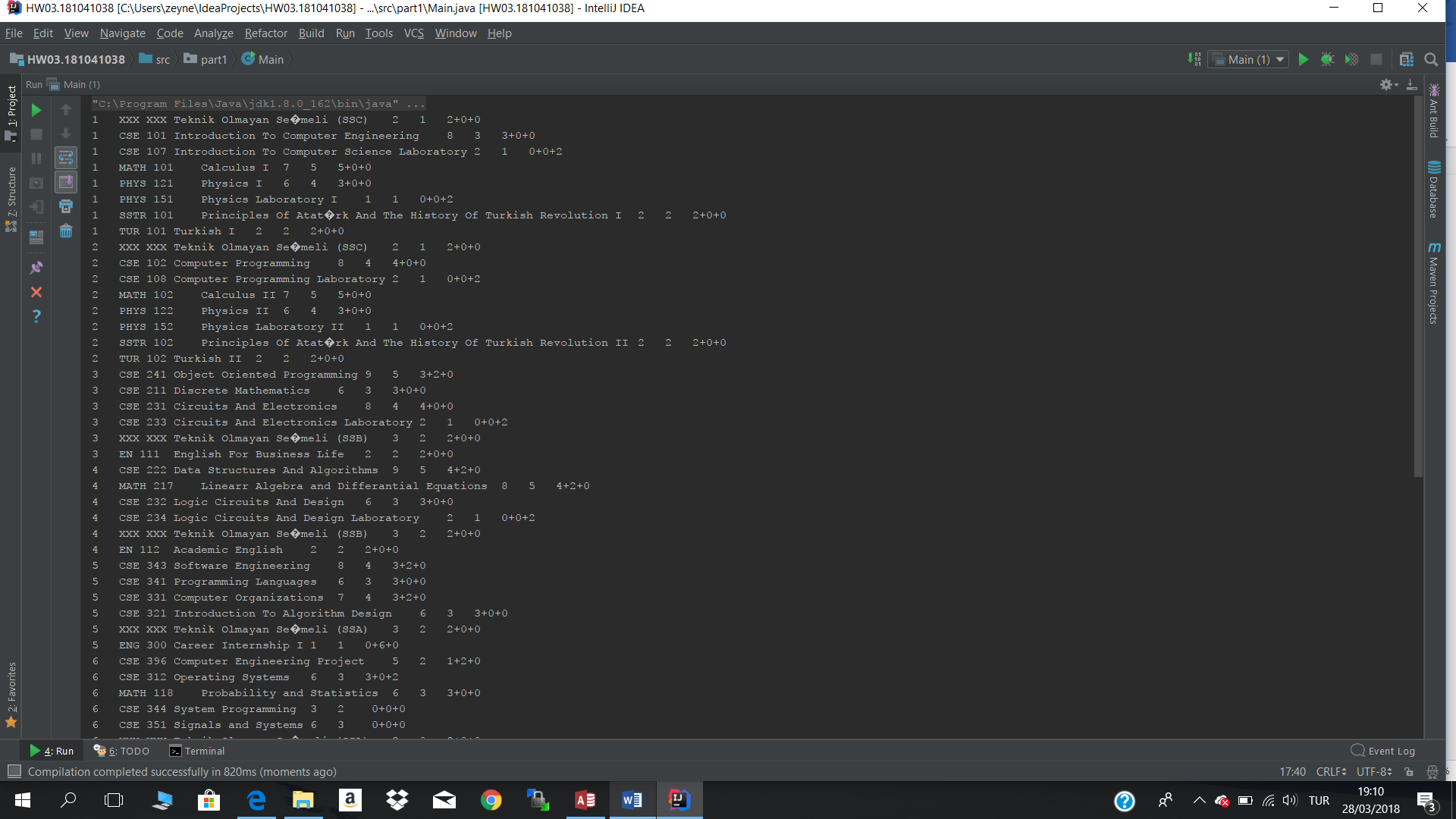
RESULT

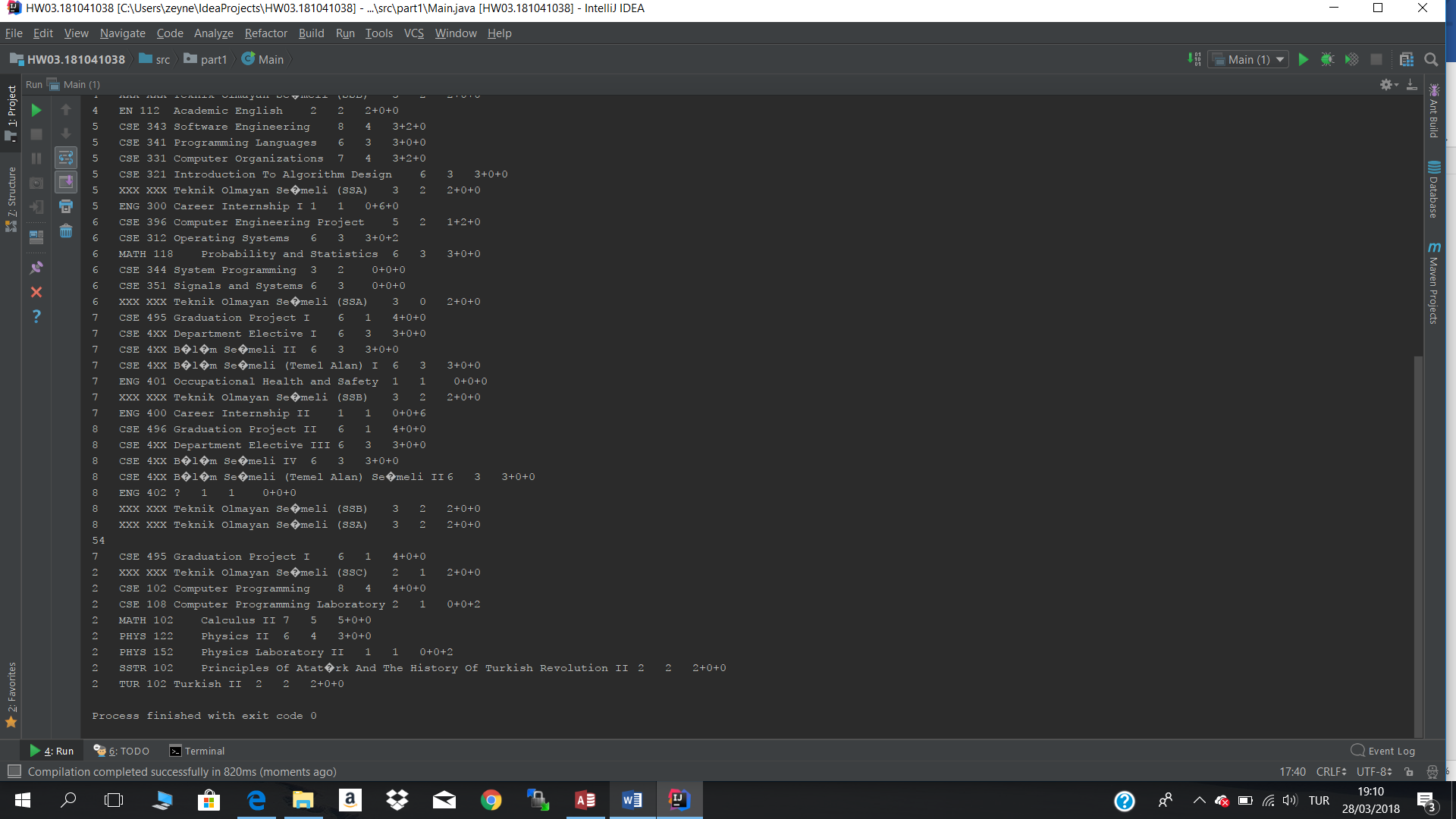
## Running Results:

For Part 1:

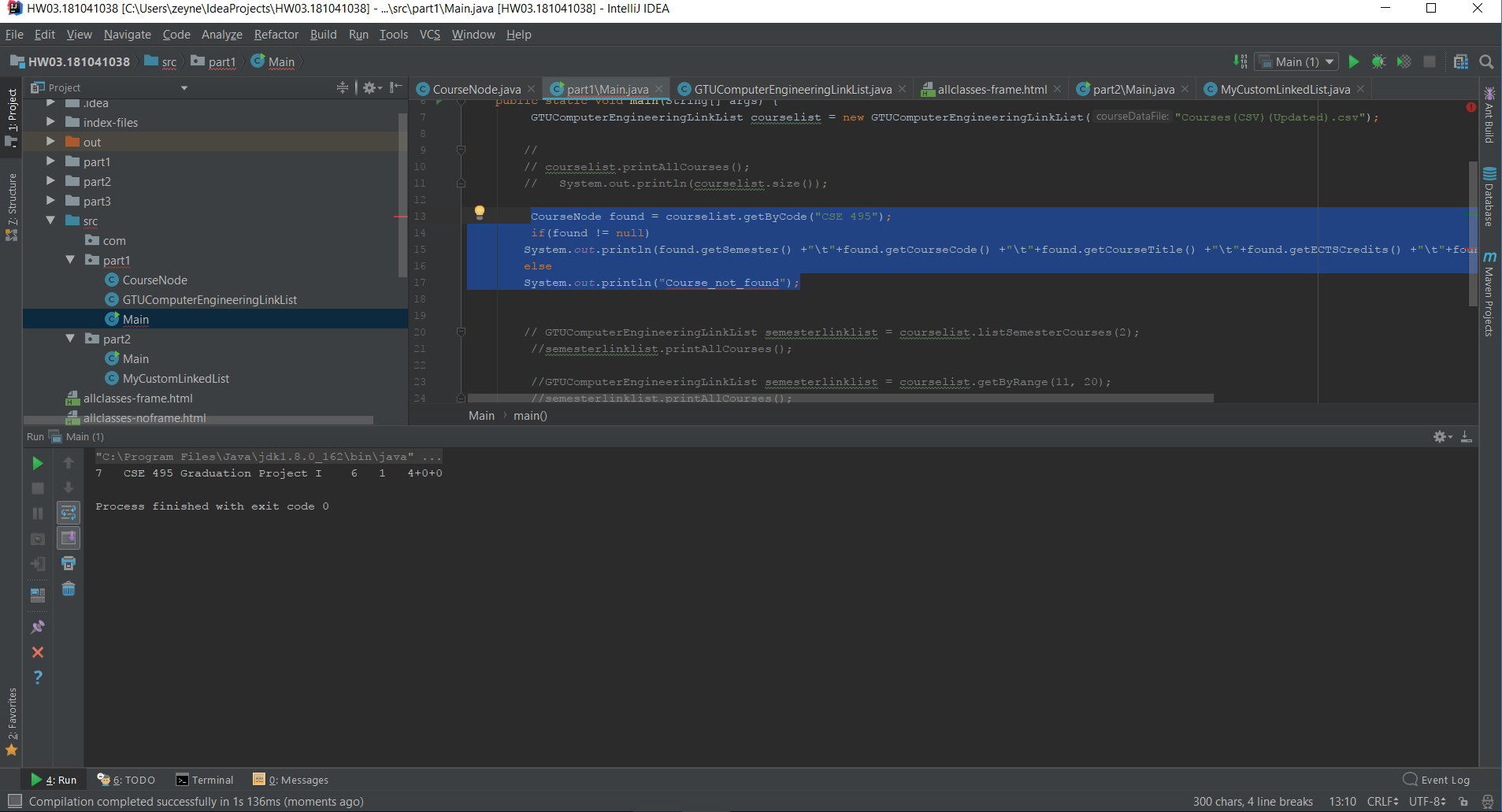
1.)

courselist.printAllCourses();  
 System.*out*.println(courselist.size());

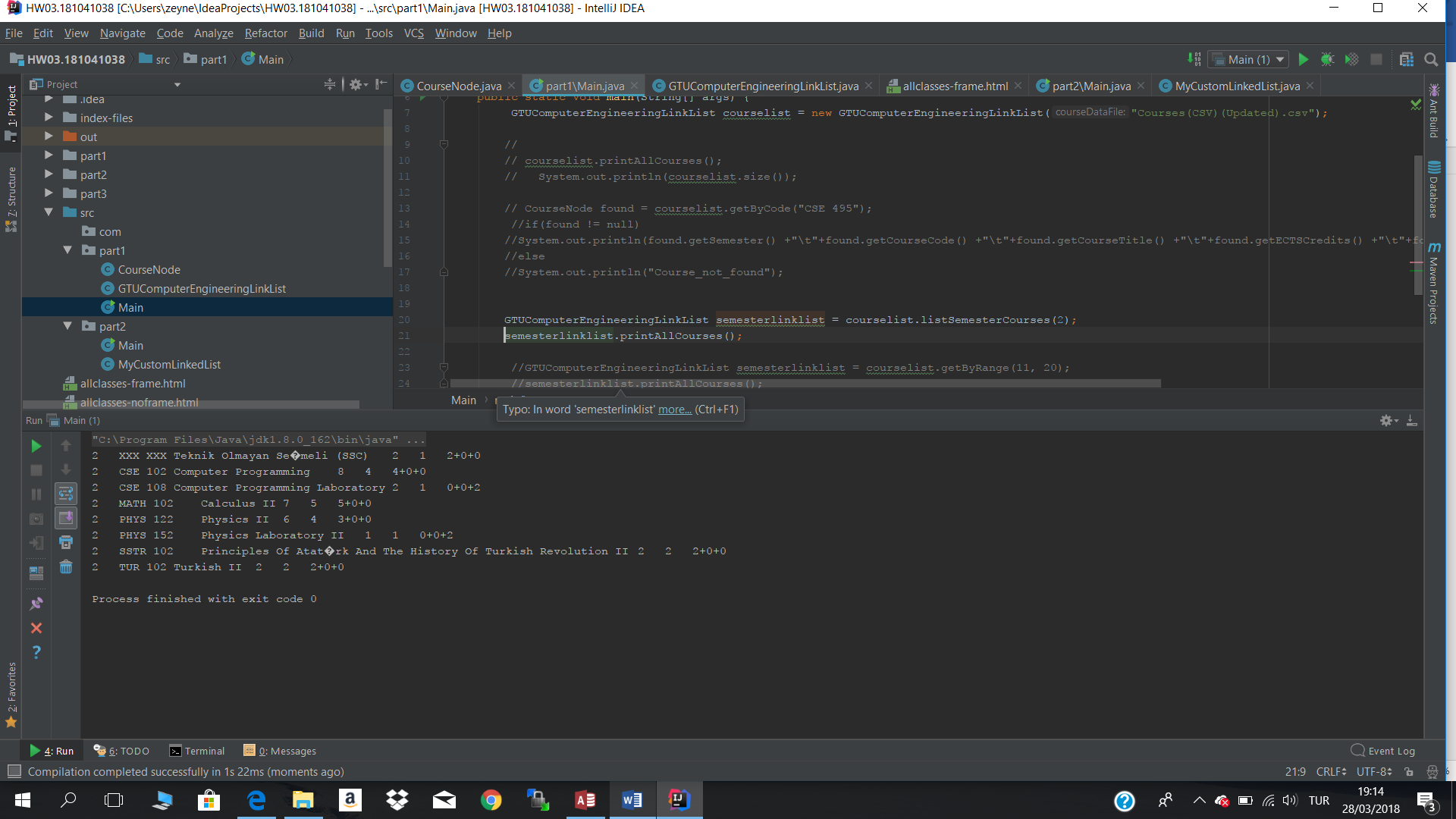




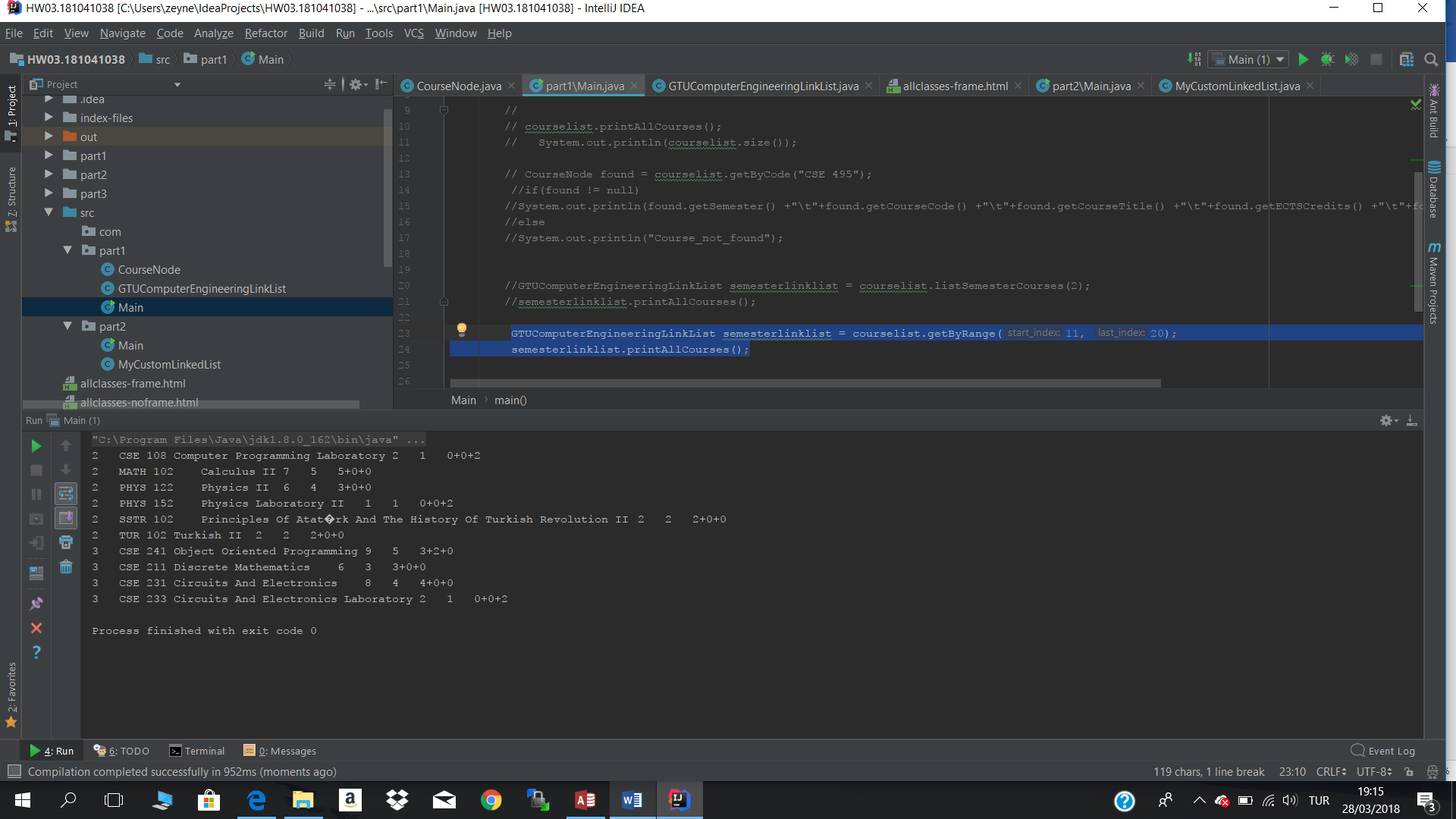
2.)



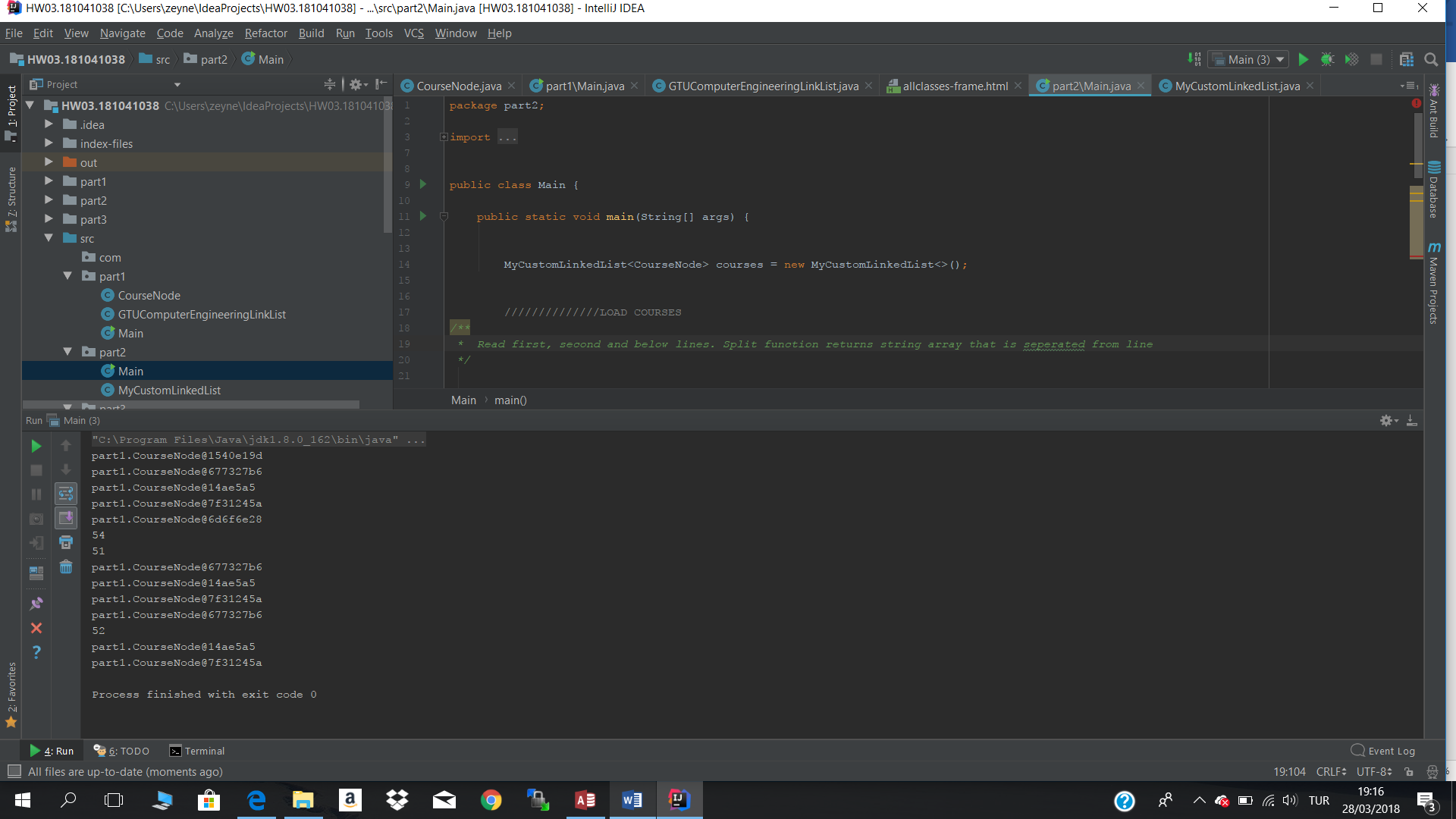
3.)



3.)



PART 2



PART 3:

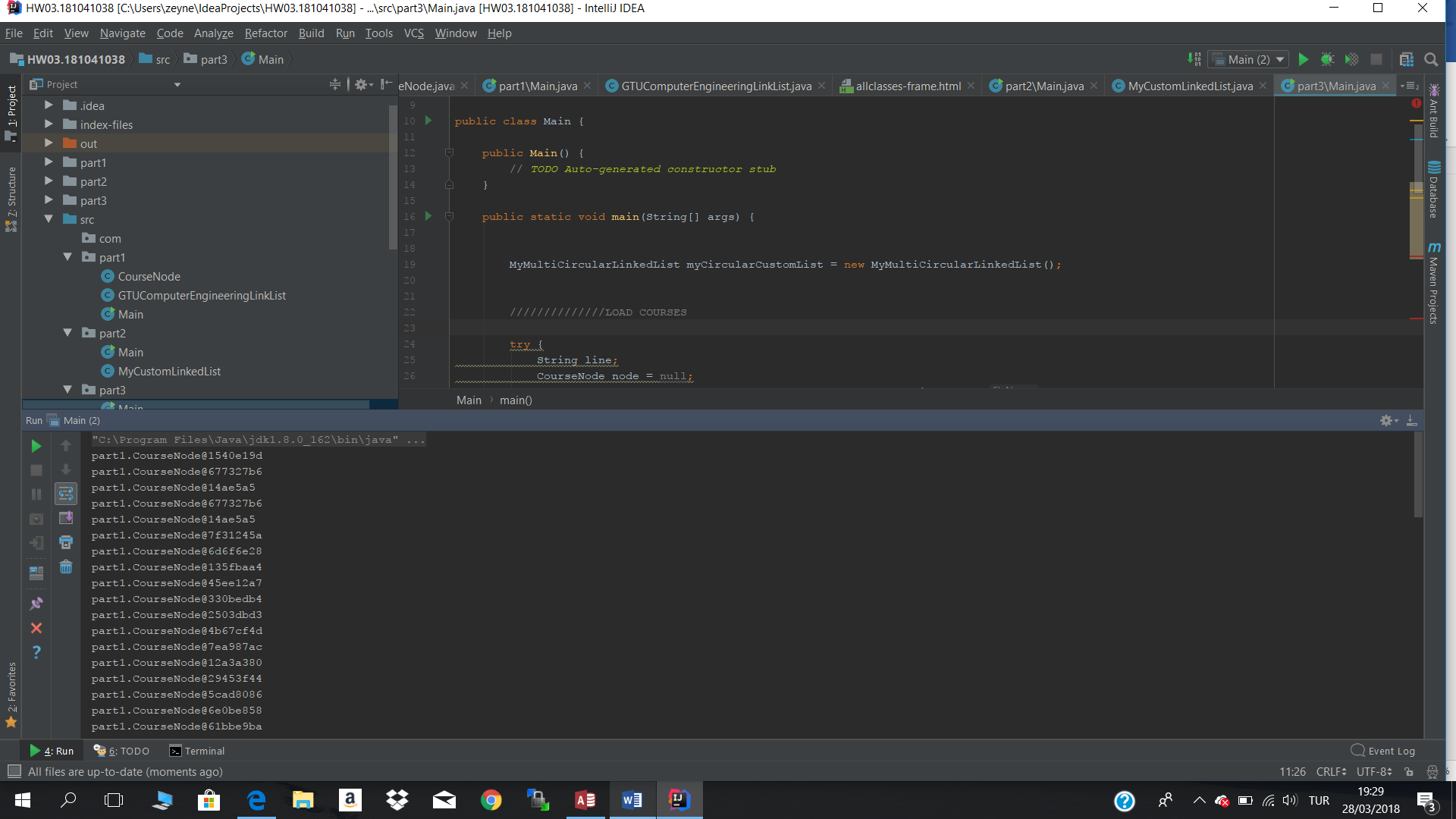
Sürekli hatalı sonuç çıkıyor neden olduğunu anlamadım. Kurslar part1.CourseNode@1540e19d

part1.CourseNode@677327b6

part1.CourseNode@14ae5a5

part1.CourseNode@7f31245a

[part1.CourseNode@6d6f6e28](mailto:part1.CourseNode@6d6f6e28) şeklinde sonuçlandı. Sürekli kodlama hatası alıyordum.



* Main titles -> 16pt , 2 line break
* Subtitles -> 14pt, 1.5 line break
* Paragraph -> 12pt, 1.5 line break