Job Sheet 16 Collection

From:

AL AZHAR RIZQI RIFA'I FIRDAUS

Class:

1 I

Absence:

01

Student Number Identity:

2241720263

Department:

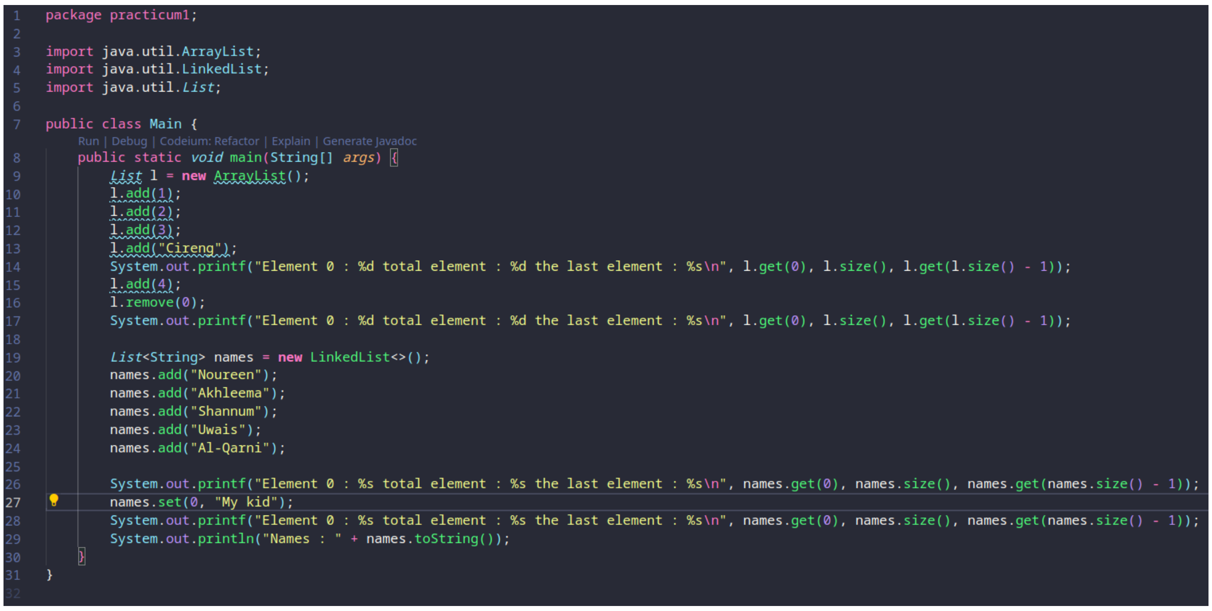
Information Technology

Study Program:

Informatics Engineering

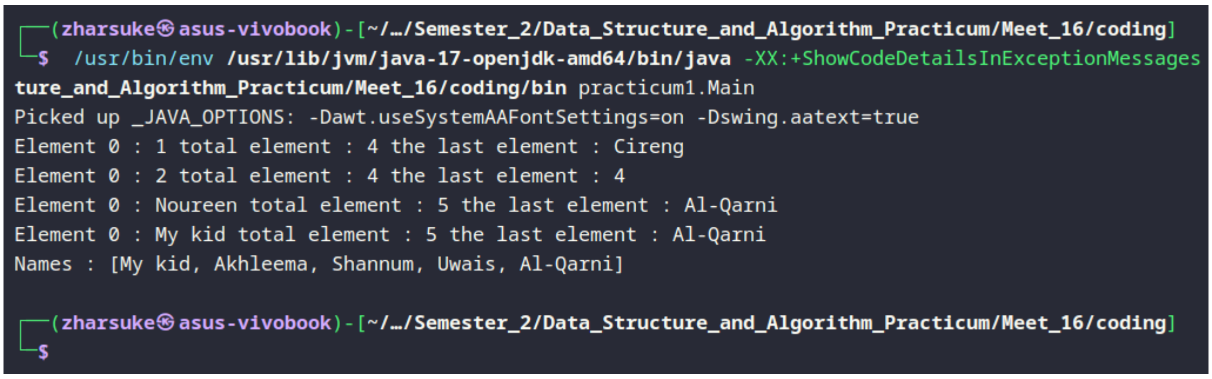
Practicum 1

Code :


Run | Debug | Codeium: Refactor | Explain | Generate Javadoc

```java
package practicum1;

import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        List l = new ArrayList();
        l.add(1);
        l.add(2);
        l.add(3);
        l.add("Cireng");
        System.out.printf("Element 0 : %d total element : %d the last element : %s\n", l.get(0), l.size(), l.get(l.size() - 1));
        l.add(4);
        l.remove(0);
        System.out.printf("Element 0 : %d total element : %d the last element : %s\n", l.get(0), l.size(), l.get(l.size() - 1));

        List<String> names = new LinkedList<>();
        names.add("Noureen");
        names.add("Akhleema");
        names.add("Shannum");
        names.add("Uwais");
        names.add("Al-Qarni");

        System.out.printf("Element 0 : %s total element : %s the last element : %s\n", names.get(0), names.size(), names.get(names.size() - 1));
        names.set(0, "My kid");
        System.out.printf("Element 0 : %s total element : %s the last element : %s\n", names.get(0), names.size(), names.get(names.size() - 1));
        System.out.println("Names : " + names.toString());
    }
}
```

Result :



```
┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$ /usr/bin/env /usr/lib/jvm/java-17-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages
ture_and_Algorithm_Practicum/Meet_16/coding/bin practicum1.Main
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Element 0 : 1 total element : 4 the last element : Cireng
Element 0 : 2 total element : 4 the last element : 4
Element 0 : Noureen total element : 5 the last element : Al-Qarni
Element 0 : My kid total element : 5 the last element : Al-Qarni
Names : [My kid, Akhleema, Shannum, Uwais, Al-Qarni]

┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$
```

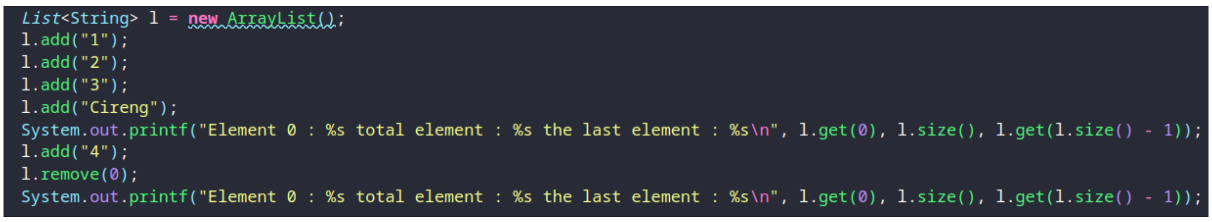
Question

1. Look at code lines 25-36, why can all types of data be accommodated in an

Arraylist?

- **Because all types of data are objects.**

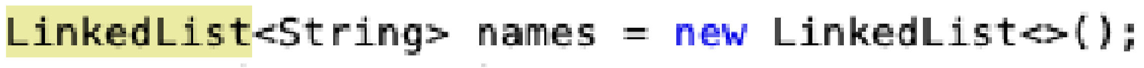
2. Modify lines of code 25-36 so that the data accommodated is only one type or a specific type of data.

- 

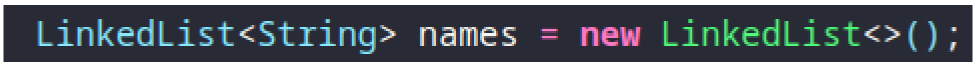
```java
List<String> l = new ArrayList();
l.add("1");
l.add("2");
l.add("3");
l.add("Cireng");
System.out.printf("Element 0 : %s total element : %s the last element : %s\n", l.get(0), l.size(), l.get(l.size() - 1));
l.add("4");
l.remove(0);
System.out.printf("Element 0 : %s total element : %s the last element : %s\n", l.get(0), l.size(), l.get(l.size() - 1));
```

- **Use Generics and set parameter type to String. Also add “” in the add method that contains a number to set in string and change string format that before %d to %s.**

3. Modify the code in line 38 to look like this



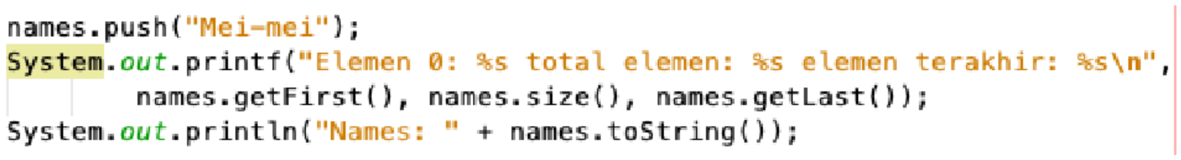
```java
LinkedList<String> names = new LinkedList<>();
```

- 

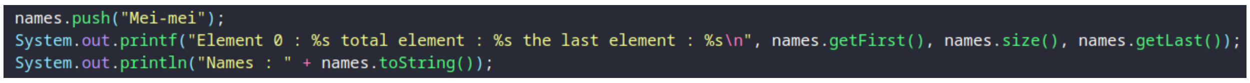
```java
LinkedList<String> names = new LinkedList<>();
```

- **The purpose we modify code above is to specify we use LinkedList implementation from interface List. We also can use LinkedList methods for example addLast, addFirst that we cannot use in List interface.**

4. Also add the following line, to give a different look from the previous one



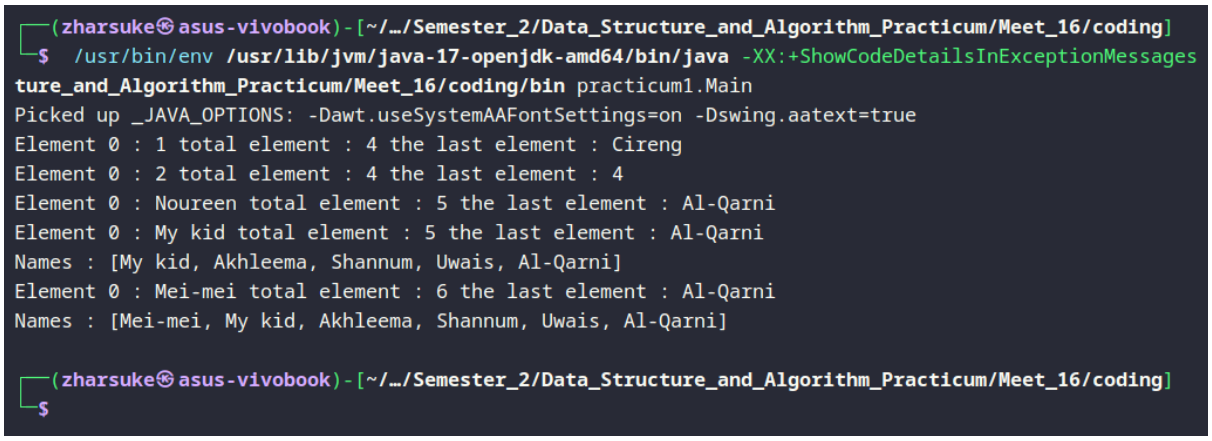
```java
names.push("Mei-mei");
System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
        names.getFirst(), names.size(), names.getLast());
System.out.println("Names: " + names.toString());
```

- 

```java
names.push("Mei-mei");
System.out.printf("Element 0 : %s total element : %s the last element : %s\n", names.getFirst(), names.size(), names.getLast());
System.out.println("Names : " + names.toString());
```

5. From the added code, please run it and what can you explain!

- Result :

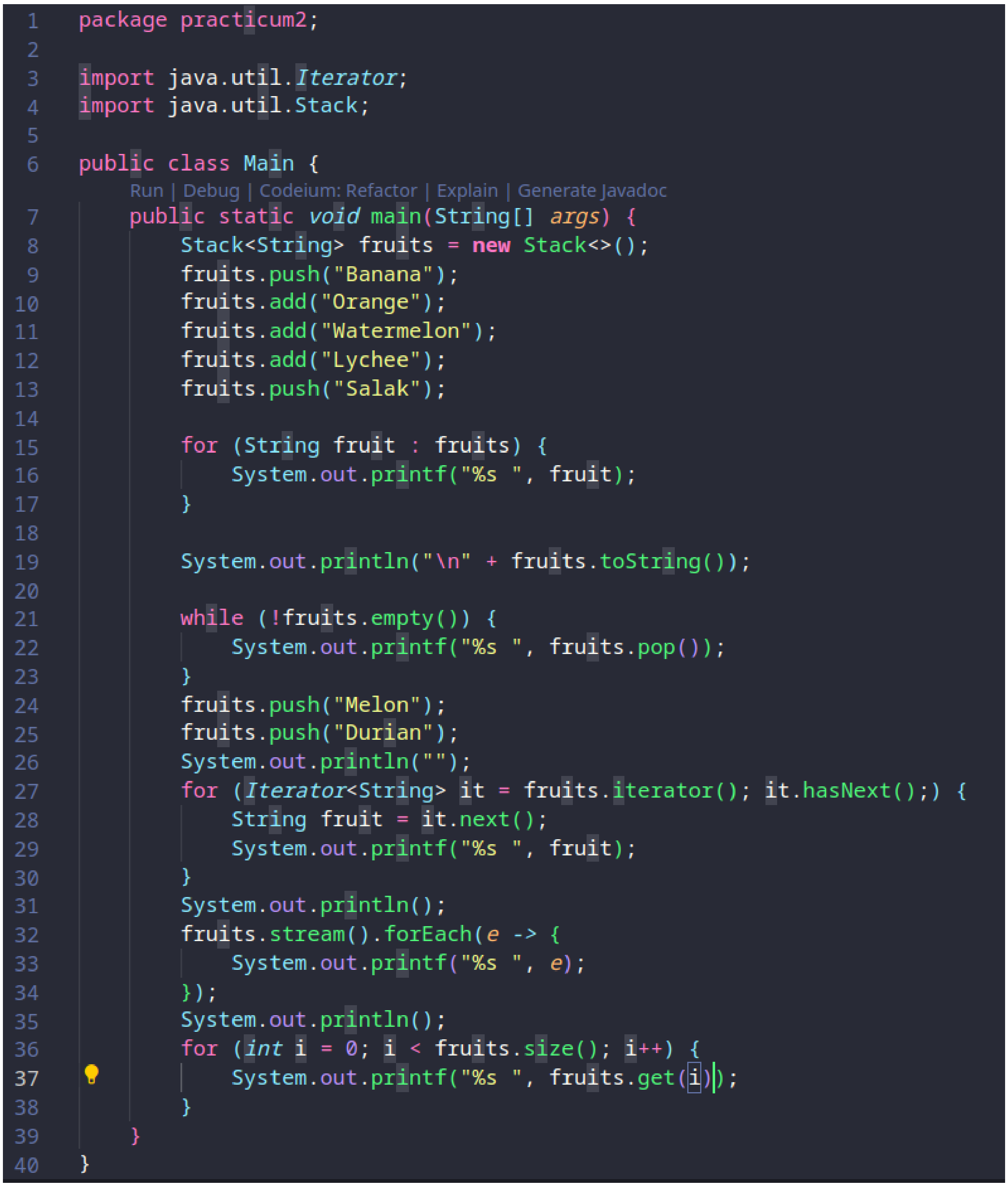


```
┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$ /usr/bin/env /usr/lib/jvm/java-17-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages
ture_and_Algorithm_Practicum/Meet_16/coding/bin practicum1.Main
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Element 0 : 1 total element : 4 the last element : Cireng
Element 0 : 2 total element : 4 the last element : 4
Element 0 : Noureen total element : 5 the last element : Al-Qarni
Element 0 : My kid total element : 5 the last element : Al-Qarni
Names : [My kid, Akhleema, Shannum, Uwais, Al-Qarni]
Element 0 : Mei-mei total element : 6 the last element : Al-Qarni
Names : [Mei-mei, My kid, Akhleema, Shannum, Uwais, Al-Qarni]

┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$
```

  - **From the code that we added, we use push, getFirst, and getLast methods from the LinkedList interface. If we change the interface to List again, the method above will make an error because in List interface there are no push, getFirst, and getLast methods.**

Practicum 2

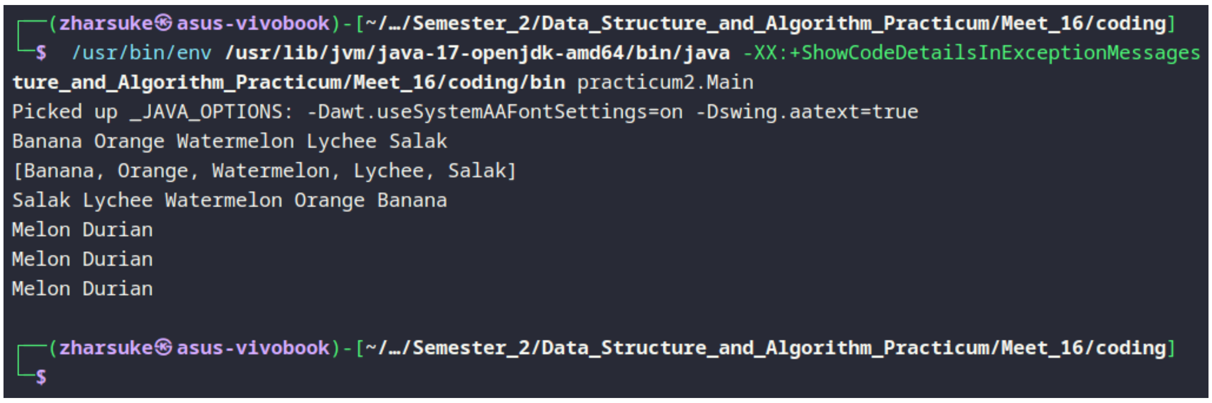
Code :



```java
package practicum2;

import java.util.Iterator;
import java.util.Stack;

public class Main {
    public static void main(String[] args) {
        Stack<String> fruits = new Stack<>();
        fruits.push("Banana");
        fruits.add("Orange");
        fruits.add("Watermelon");
        fruits.add("Lychee");
        fruits.push("Salak");

        for (String fruit : fruits) {
            System.out.printf("%s ", fruit);
        }

        System.out.println("\n" + fruits.toString());

        while (!fruits.empty()) {
            System.out.printf("%s ", fruits.pop());
        }
        fruits.push("Melon");
        fruits.push("Durian");
        System.out.println("");
        for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
            String fruit = it.next();
            System.out.printf("%s ", fruit);
        }
        System.out.println();
        fruits.stream().forEach(e -> {
            System.out.printf("%s ", e);
        });
        System.out.println();
        for (int i = 0; i < fruits.size(); i++) {
            System.out.printf("%s ", fruits.get(i));
        }
    }
}
```

Run | Debug | Codeium: Refactor | Explain | Generate Javadoc

Result :



```
┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$  /usr/bin/env /usr/lib/jvm/java-17-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages
ture_and_Algorithm_Practicum/Meet_16/coding/bin practicum2.Main
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Banana Orange Watermelon Lychee Salak
[Banana, Orange, Watermelon, Lychee, Salak]
Salak Lychee Watermelon Orange Banana
Melon Durian
Melon Durian
Melon Durian

┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$
```

Questions

1. What is the difference between the push() and add() functions on the fruits object?

- **The difference between both is the push() method contained in stack and the add() method contained in List interface. We can use push() and add() methods in stack, but can't use push() method in List because push() methods have specific behavior to stack, then not like add() method that have general behavior. Both have the same function to add elements.**

2. Please remove lines 43 and 44, what will happen? Why is that?

- **The iteration of melon and durian doesn't appear because we remove them. The reason why only melon and durian that appear before we remove them is because there is an iteration that pop all of the elements. The function of the pop method is to remove elements.**

3. Explain the function of lines 46-49?

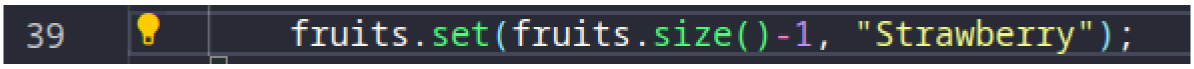
- **The function is to iterate over a collection using an iterator. The first line, creates an object from Iterator<> class that implements iterator() that allows us to traverse the elements one by one. Then hasNext() method checks if there are more elements in the collection. next() method to retrieve the next elements of collections.**

4. Please change line 25, Stack<String> to List<String> and what happens?

Why is this possible?

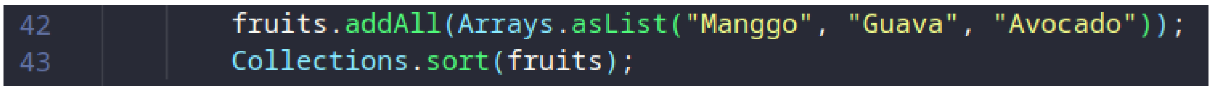
- **After modification, the push(), empty(), and pop() method is undefined for List<> because they are contained in stack, not in List.**

5. Change the last element of the fruits object to "Strawberry"!

- 

```
39        fruits.set(fruits.size()-1, "Strawberry");
```

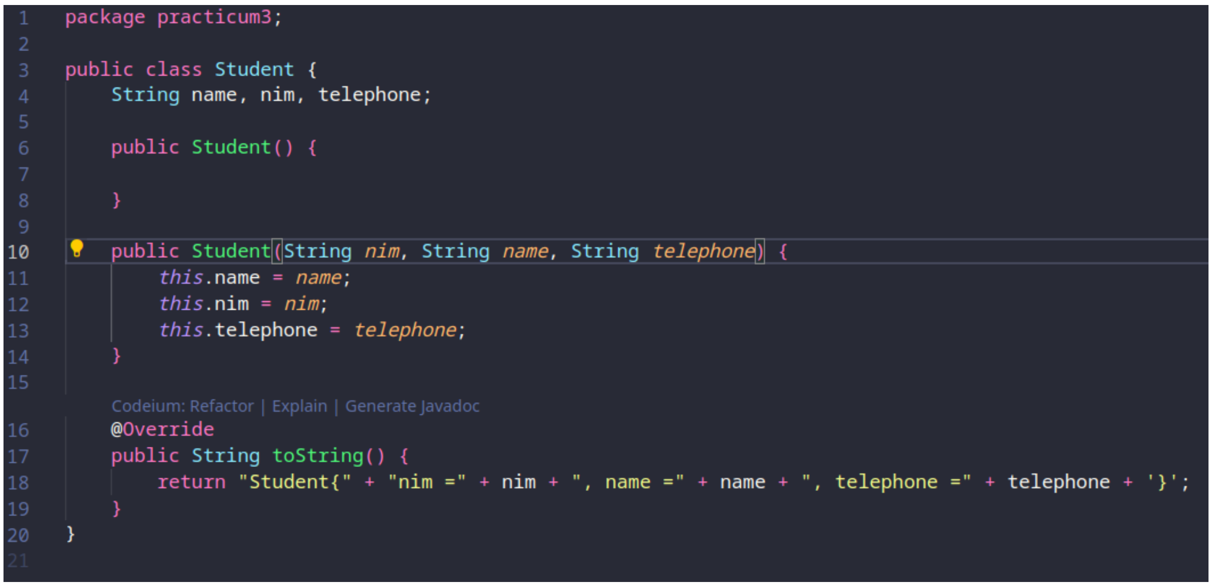
6. Add 3 fruits such as "Mango", "guava", and "avocado" and sort them!

- 

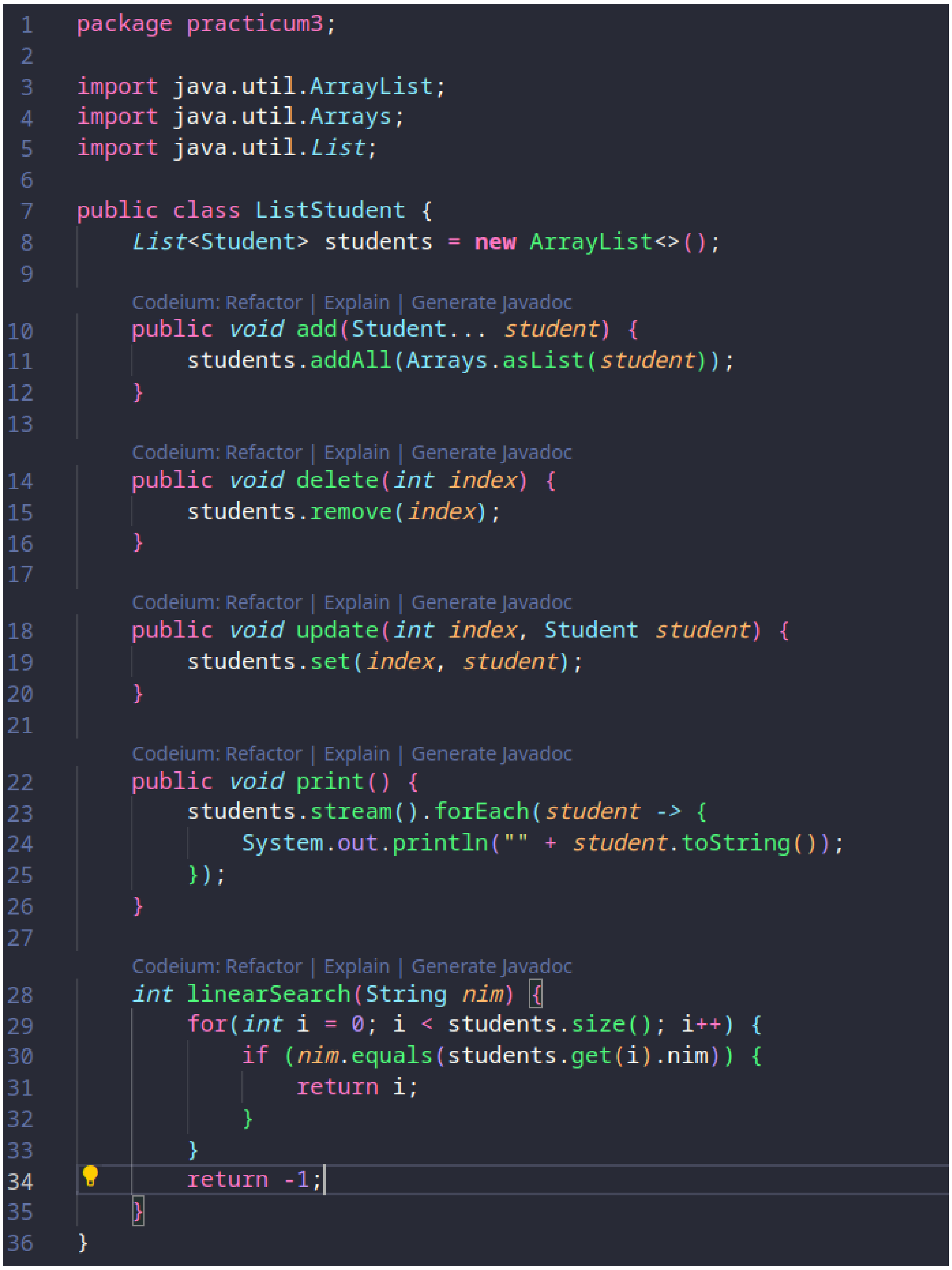
```
42        fruits.addAll(Arrays.asList("Manggo", "Guava", "Avocado"));
43        Collections.sort(fruits);
```

Practicum 3

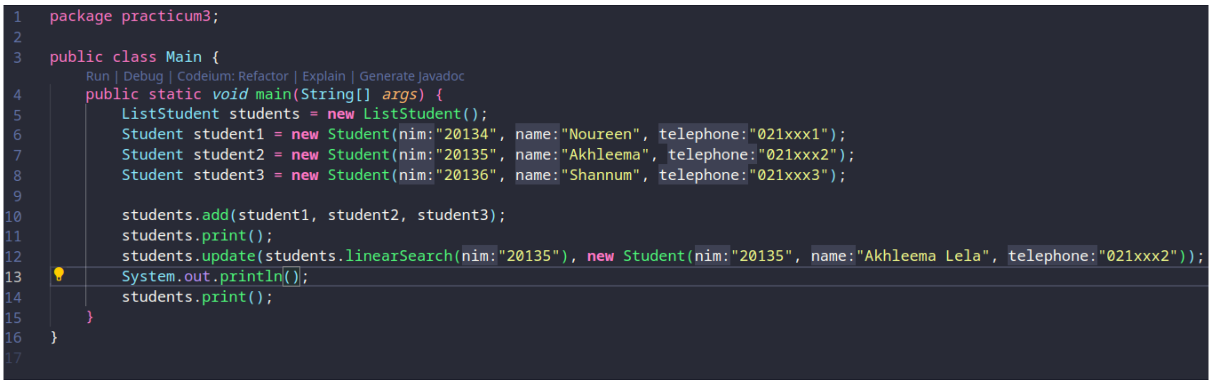
Code :



```java
package practicum3;

public class Student {
    String name, nim, telephone;

    public Student() {

    }

    public Student(String nim, String name, String telephone) {
        this.name = name;
        this.nim = nim;
        this.telephone = telephone;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    @Override
    public String toString() {
        return "Student{" + "nim =" + nim + ", name =" + name + ", telephone =" + telephone + '}';
    }
}
```

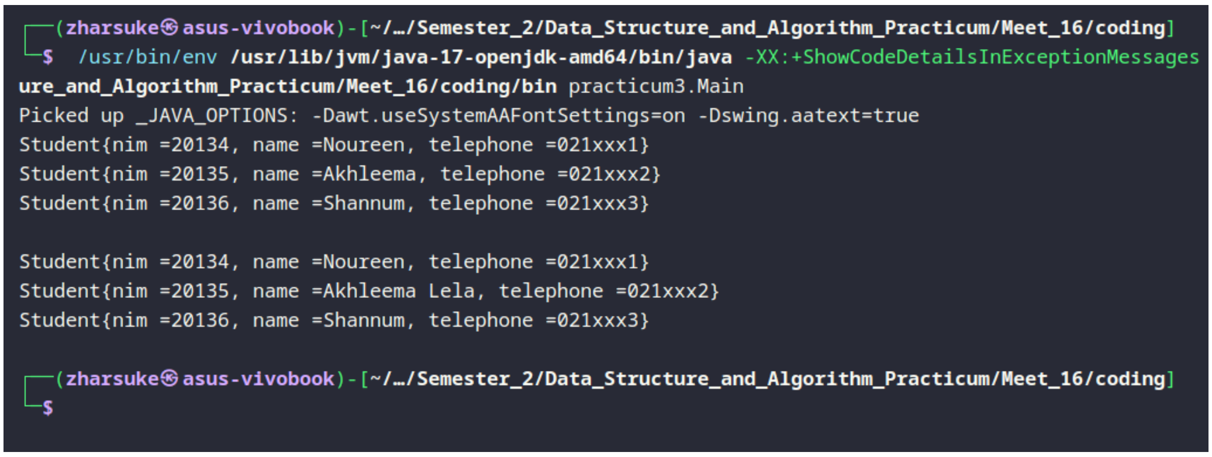

Codeium: Refactor | Explain | Generate Javadoc
Codeium: Refactor | Explain | Generate Javadoc
Codeium: Refactor | Explain | Generate Javadoc
Codeium: Refactor | Explain | Generate Javadoc
Codeium: Refactor | Explain | Generate Javadoc

```java
package practicum3;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class ListStudent {
    List<Student> students = new ArrayList<>();

    public void add(Student... student) {
        students.addAll(Arrays.asList(student));
    }

    public void delete(int index) {
        students.remove(index);
    }

    public void update(int index, Student student) {
        students.set(index, student);
    }

    public void print() {
        students.stream().forEach(student -> {
            System.out.println("" + student.toString());
        });
    }

    int linearSearch(String nim) {
        for(int i = 0; i < students.size(); i++) {
            if (nim.equals(students.get(i).nim)) {
                return i;
            }
        }
        return -1;
    }
}
```



```java
package practicum3;

public class Main {
    public static void main(String[] args) {
        ListStudent students = new ListStudent();
        Student student1 = new Student(nim:"20134", name:"Noureen", telephone:"021xxx1");
        Student student2 = new Student(nim:"20135", name:"Akhleema", telephone:"021xxx2");
        Student student3 = new Student(nim:"20136", name:"Shannum", telephone:"021xxx3");

        students.add(student1, student2, student3);
        students.print();
        students.update(students.linearSearch(nim:"20135"), new Student(nim:"20135", name:"Akhleema Lela", telephone:"021xxx2"));
        System.out.println();
        students.print();
    }
}
```

Result :



```
┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$ /usr/bin/env /usr/lib/jvm/java-17-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages
ure_and_Algorithm_Practicum/Meet_16/coding/bin practicum3.Main
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Student{nim =20134, name =Noureen, telephone =021xxx1}
Student{nim =20135, name =Akhleema, telephone =021xxx2}
Student{nim =20136, name =Shannum, telephone =021xxx3}

Student{nim =20134, name =Noureen, telephone =021xxx1}
Student{nim =20135, name =Akhleema Lela, telephone =021xxx2}
Student{nim =20136, name =Shannum, telephone =021xxx3}

┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$
```

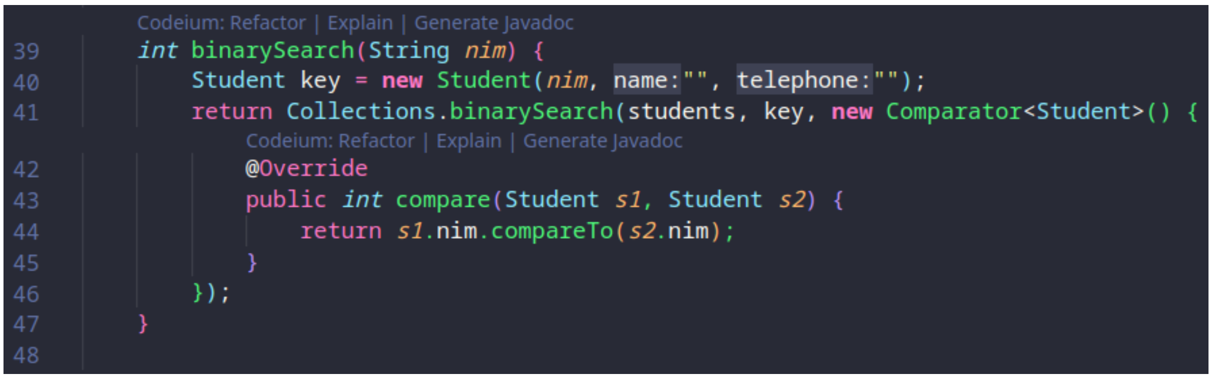
Questions

1. In the add() function that uses unlimited arguments, what concept is used?

And what are its advantages?

- **It used varargs parameters. The varargs parameter allows methods to be more flexible in terms of accepting a variable number of arguments of a specified type.**

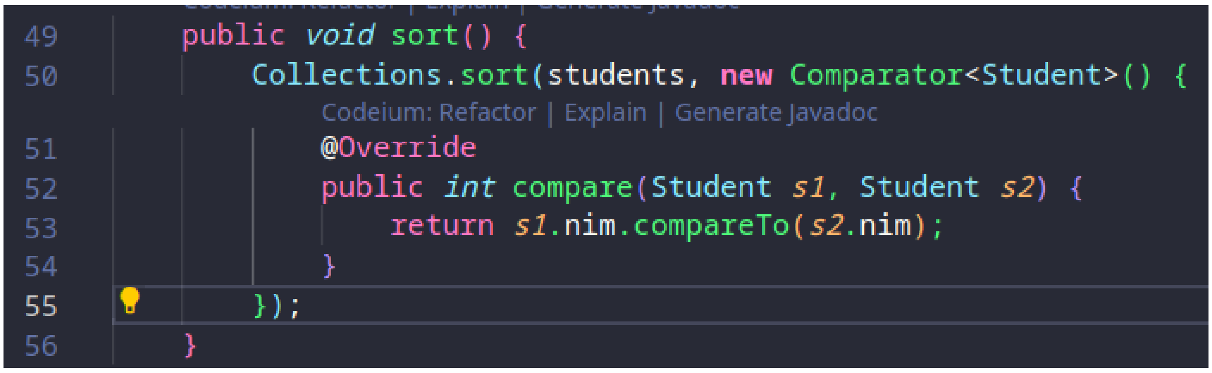
2. In the linearSearch() function above, please replace it with the binarySearch() function from the collection!

- 

```java
    int binarySearch(String nim) {
        Student key = new Student(nim, name:"", telephone:"");
        return Collections.binarySearch(students, key, new Comparator<Student>() {
            @Override
            public int compare(Student s1, Student s2) {
                return s1.nim.compareTo(s2.nim);
            }
        });
    }
```

3. Add an ascending or descending sorting function to the class!

-



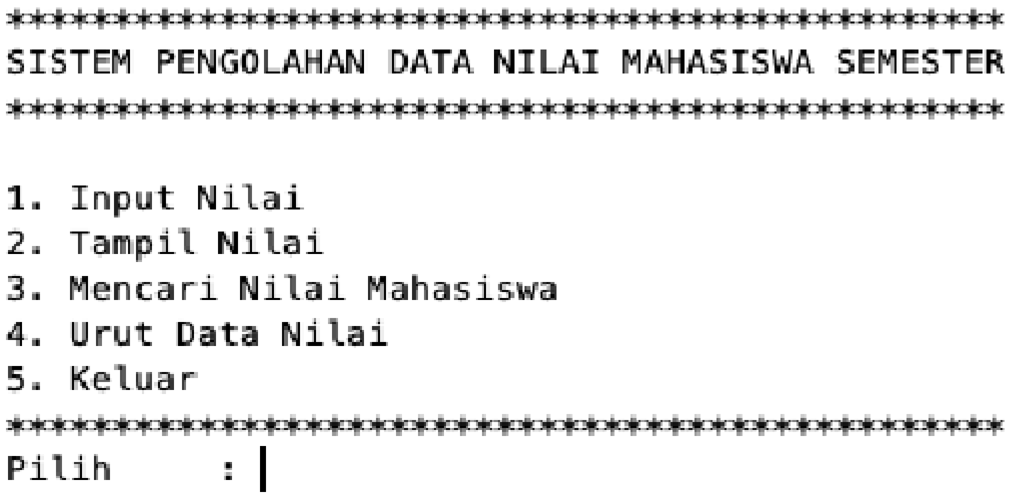
```java
public void sort() {
    Collections.sort(students, new Comparator<Student>() {
        Codeium: Refactor | Explain | Generate Javadoc
        @Override
        public int compare(Student s1, Student s2) {
            return s1.nim.compareTo(s2.nim);
        }
    });
}
```

Assignment

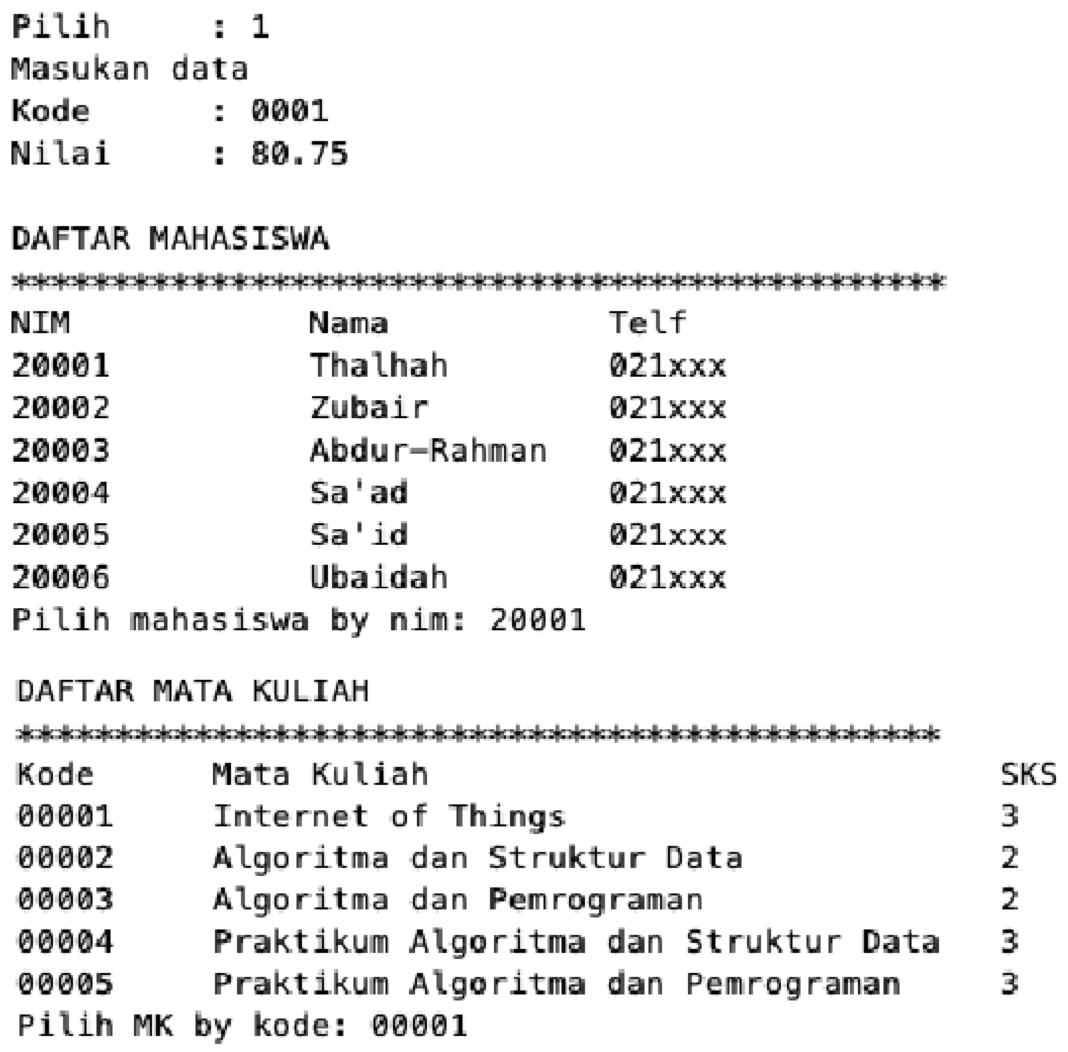
1. Implement a semester student grade list program, which has at least 3 classes, namely Student, Grade, and Course. Student and Course data need to go through data inputting data first.

Program Illustration

Start Menu and Data Addition

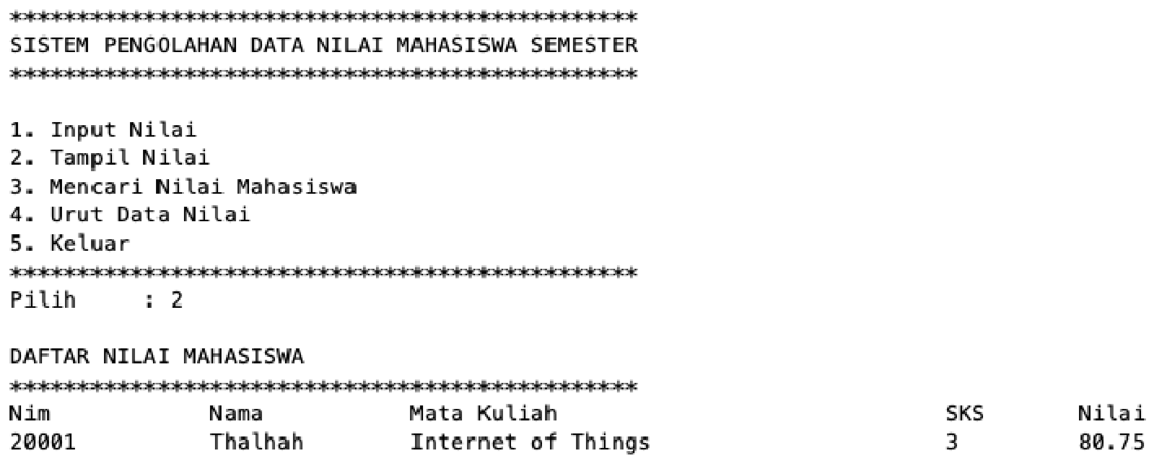


```
***************************************************
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
***************************************************

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
***************************************************
Pilih      :
```



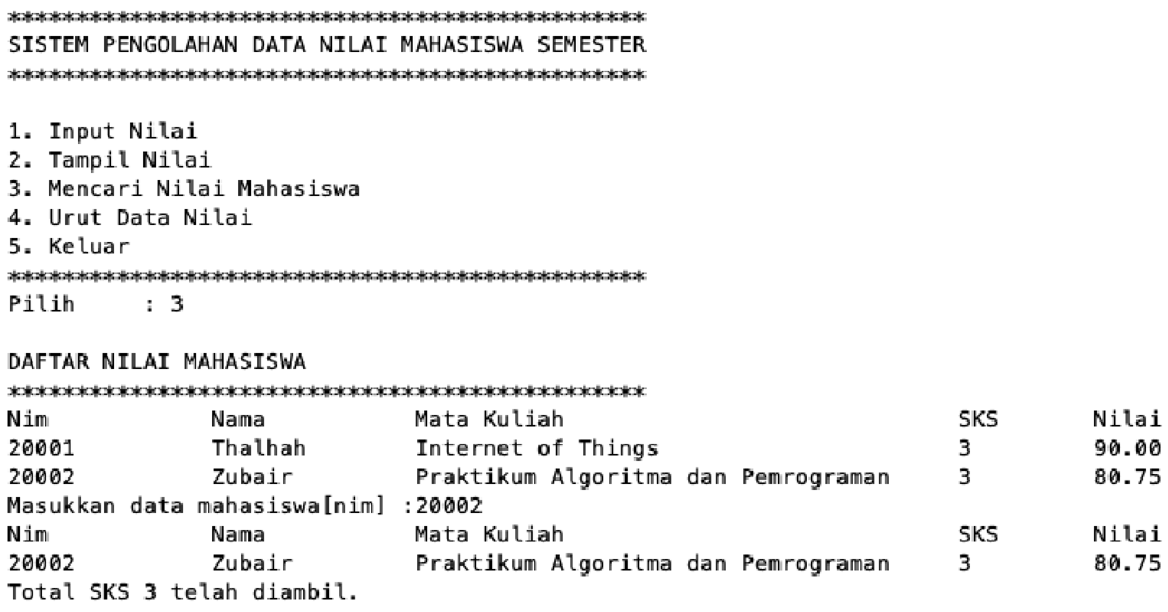
```
Pilih     : 1
Masukan data
Kode      : 0001
Nilai     : 80.75

DAFTAR MAHASISWA
************************************************
NIM            Nama           Telf
20001          Thalhah        021xxx
20002          Zubair         021xxx
20003          Abdur-Rahman   021xxx
20004          Sa'ad          021xxx
20005          Sa'id          021xxx
20006          Ubaidah        021xxx
Pilih mahasiswa by nim: 20001

DAFTAR MATA KULIAH
************************************************
Kode      Mata Kuliah                             SKS
00001     Internet of Things                      3
00002     Algoritma dan Struktur Data             2
00003     Algoritma dan Pemrograman               2
00004     Praktikum Algoritma dan Struktur Data   3
00005     Praktikum Algoritma dan Pemrograman     3
Pilih MK by kode: 00001
```

Display Value



```
************************************************
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
************************************************

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
************************************************
Pilih     : 2

DAFTAR NILAI MAHASISWA
************************************************
Nim          Nama         Mata Kuliah                             SKS       Nilai
20001        Thalhah      Internet of Things                      3         80.75
```

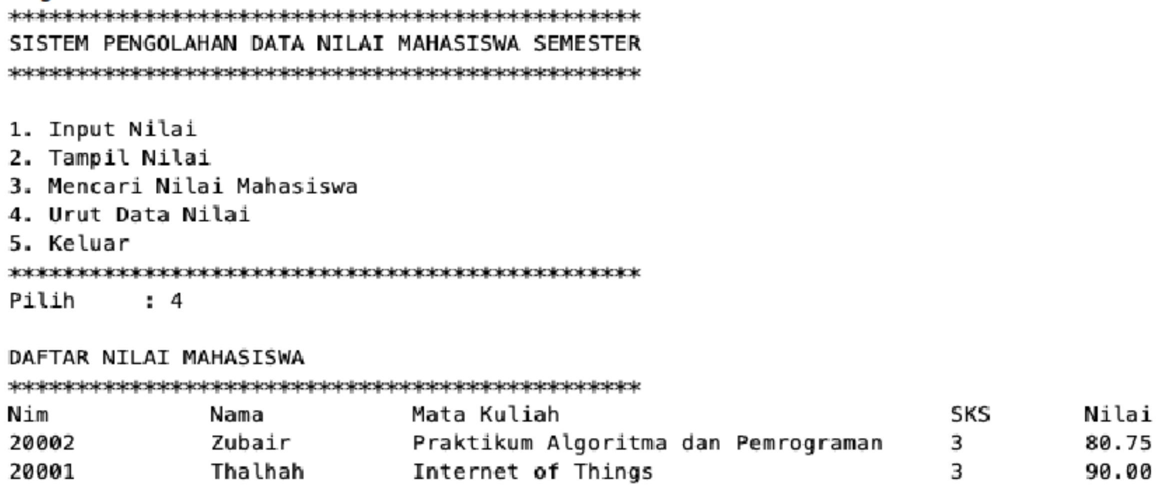
Student Data Search



```
************************************************
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
************************************************

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
************************************************
Pilih     : 3

DAFTAR NILAI MAHASISWA
************************************************
Nim            Nama           Mata Kuliah                         SKS       Nilai
20001          Thalhah        Internet of Things                  3         90.00
20002          Zubair         Praktikum Algoritma dan Pemrograman 3         80.75
Masukkan data mahasiswa[nim] :20002
Nim            Nama           Mata Kuliah                         SKS       Nilai
20002          Zubair         Praktikum Algoritma dan Pemrograman 3         80.75
Total SKS 3 telah diambil.
```

Sorting of Value Data

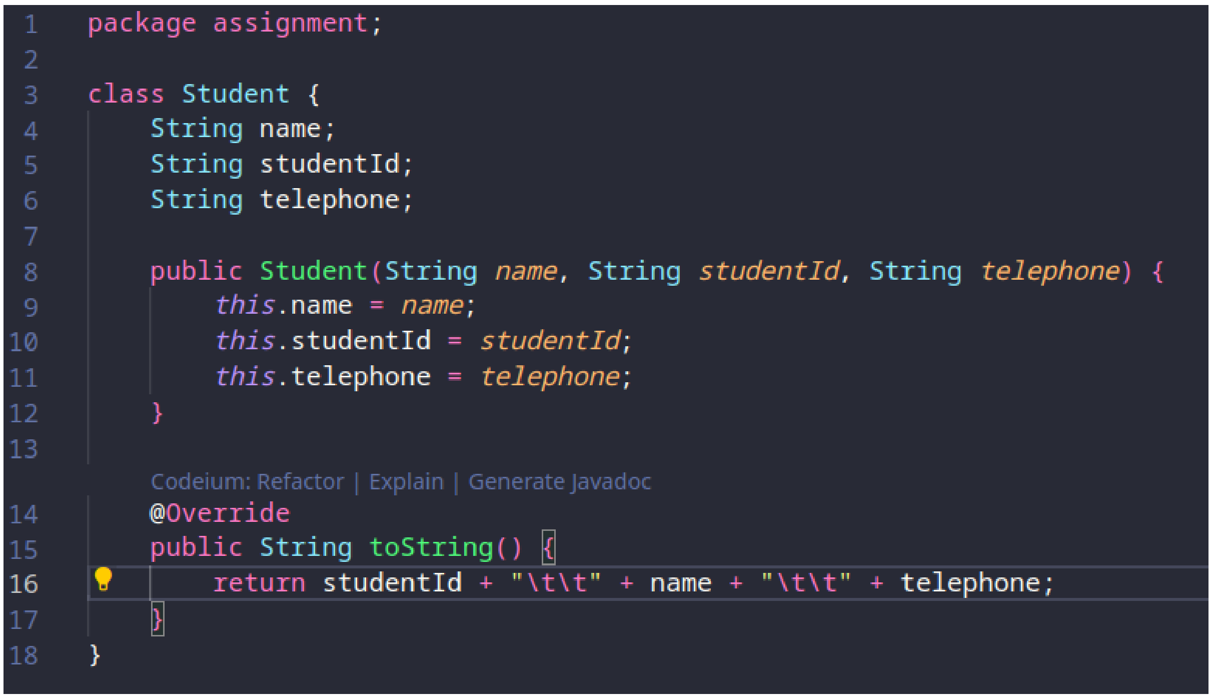
Pengurutan Data Nilai



```
************************************************
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
************************************************

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
************************************************
Pilih     : 4

DAFTAR NILAI MAHASISWA
************************************************
Nim            Nama           Mata Kuliah                         SKS       Nilai
20002          Zubair         Praktikum Algoritma dan Pemrograman 3         80.75
20001          Thalhah        Internet of Things                  3         90.00
```

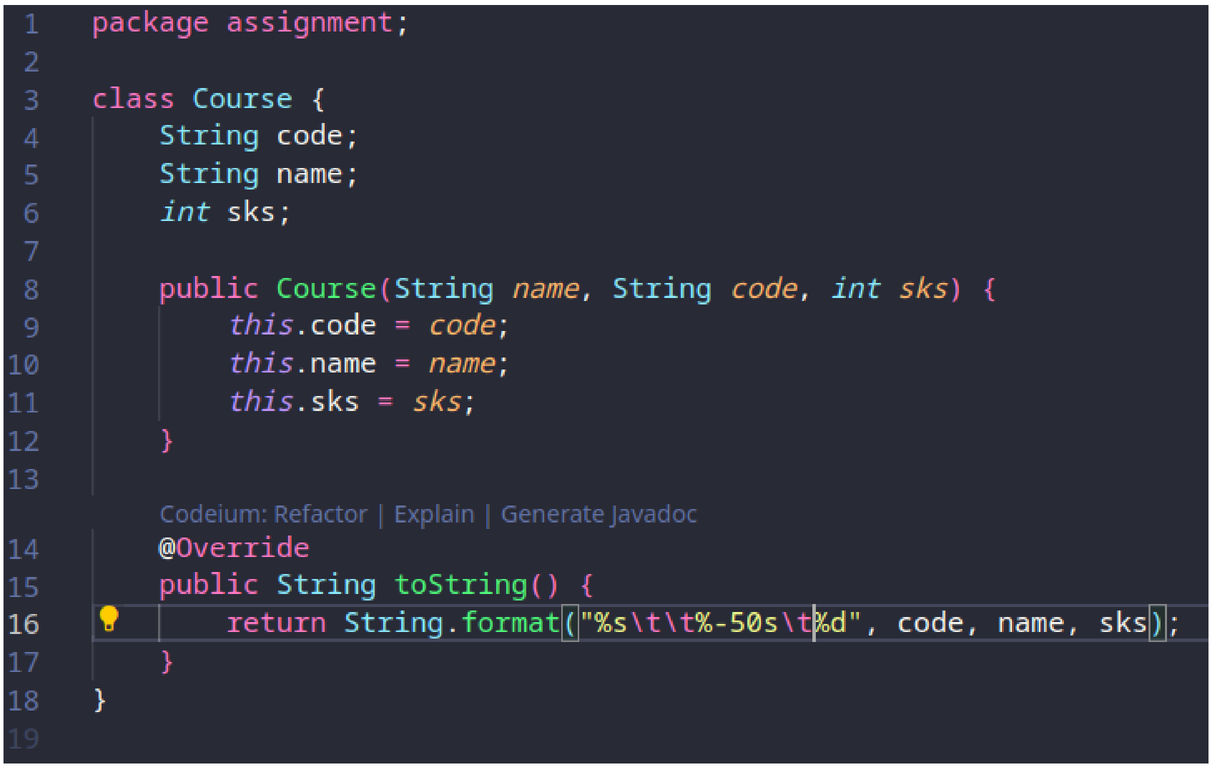
- Code :

  Student



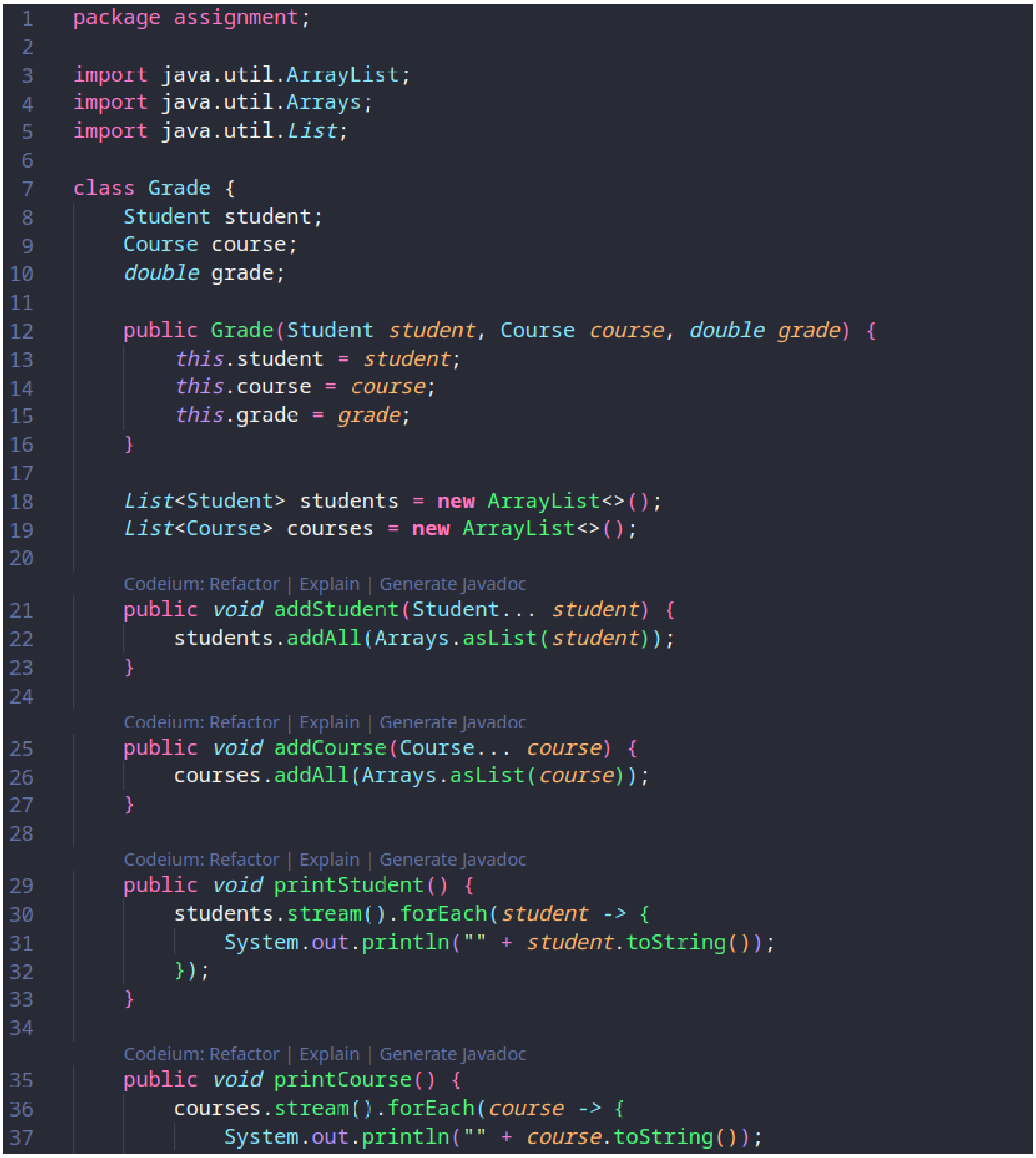
```java
package assignment;

class Student {
    String name;
    String studentId;
    String telephone;

    public Student(String name, String studentId, String telephone) {
        this.name = name;
        this.studentId = studentId;
        this.telephone = telephone;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    @Override
    public String toString() {
        return studentId + "\t\t" + name + "\t\t" + telephone;
    }
}
```

Course

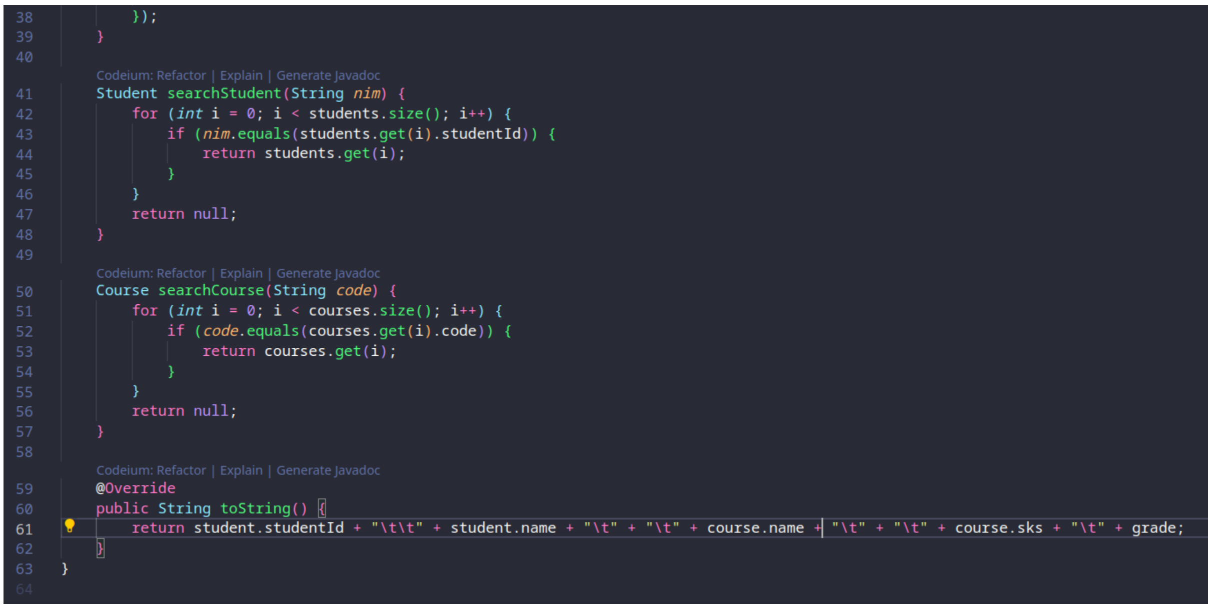


```java
package assignment;

class Course {
    String code;
    String name;
    int sks;

    public Course(String name, String code, int sks) {
        this.code = code;
        this.name = name;
        this.sks = sks;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    @Override
    public String toString() {
        return String.format("%s\t\t%-50s\t%d", code, name, sks);
    }
}
```

Grade

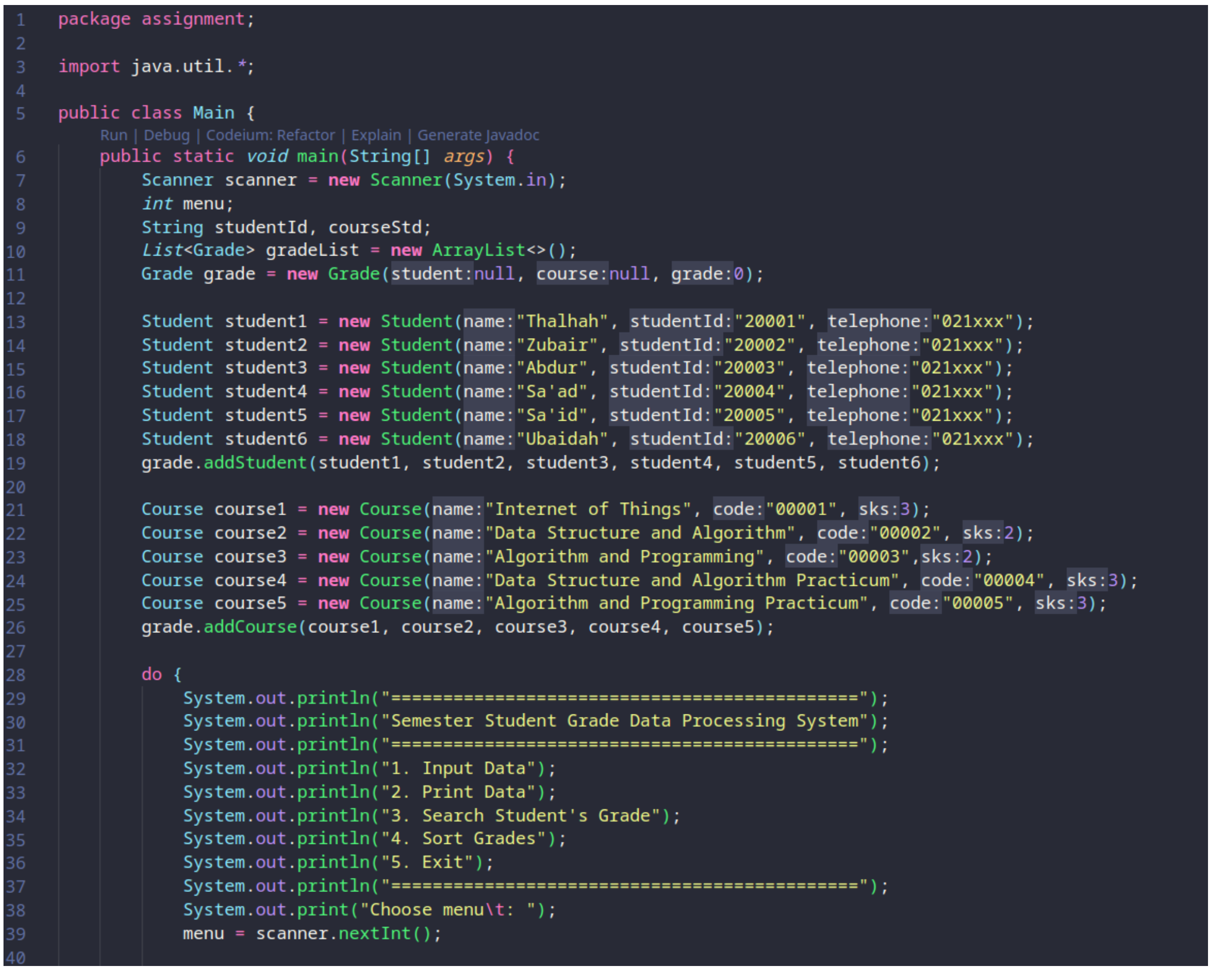


```java
package assignment;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

class Grade {
    Student student;
    Course course;
    double grade;

    public Grade(Student student, Course course, double grade) {
        this.student = student;
        this.course = course;
        this.grade = grade;
    }

    List<Student> students = new ArrayList<>();
    List<Course> courses = new ArrayList<>();

    Codeium: Refactor | Explain | Generate Javadoc
    public void addStudent(Student... student) {
        students.addAll(Arrays.asList(student));
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void addCourse(Course... course) {
        courses.addAll(Arrays.asList(course));
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void printStudent() {
        students.stream().forEach(student -> {
            System.out.println("" + student.toString());
        });
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void printCourse() {
        courses.stream().forEach(course -> {
            System.out.println("" + course.toString());
```

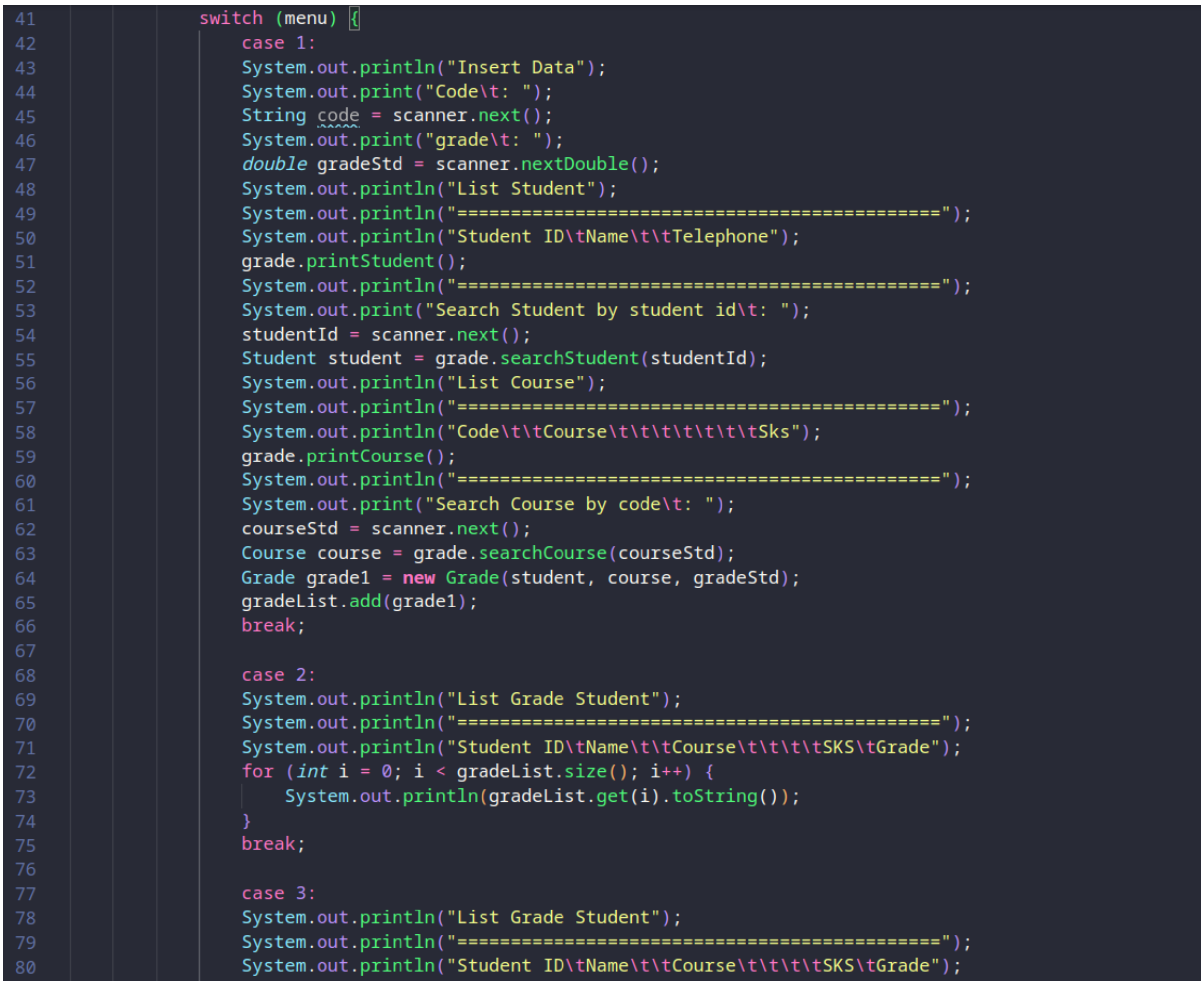


```
        });
    }

    Codeium: Refactor | Explain | Generate Javadoc
    Student searchStudent(String nim) {
        for (int i = 0; i < students.size(); i++) {
            if (nim.equals(students.get(i).studentId)) {
                return students.get(i);
            }
        }
        return null;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    Course searchCourse(String code) {
        for (int i = 0; i < courses.size(); i++) {
            if (code.equals(courses.get(i).code)) {
                return courses.get(i);
            }
        }
        return null;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    @Override
    public String toString() {
        return student.studentId + "\t\t" + student.name + "\t" + "\t" + course.name + "\t" + "\t" + course.sks + "\t" + grade;
    }
}
```

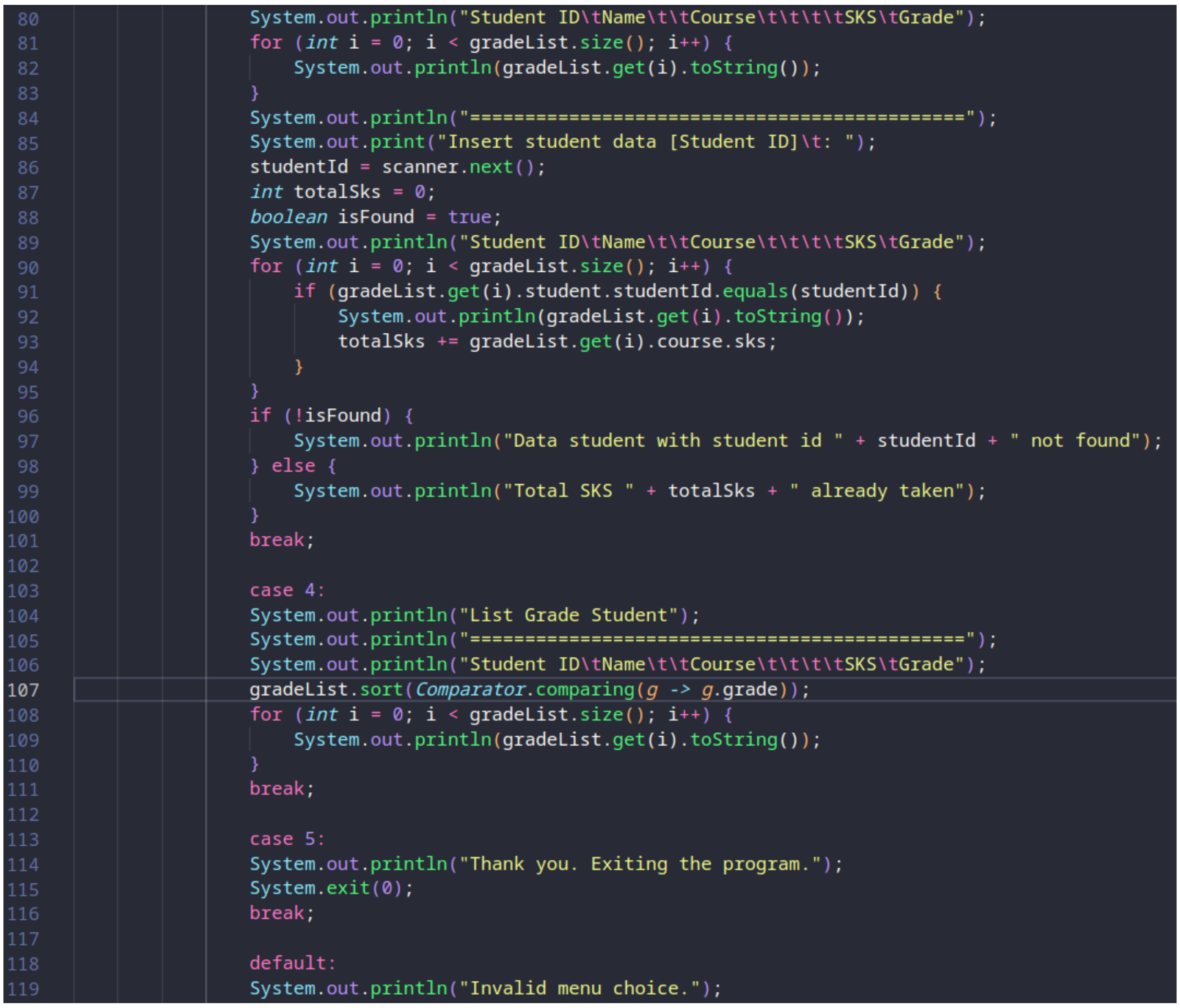
Main



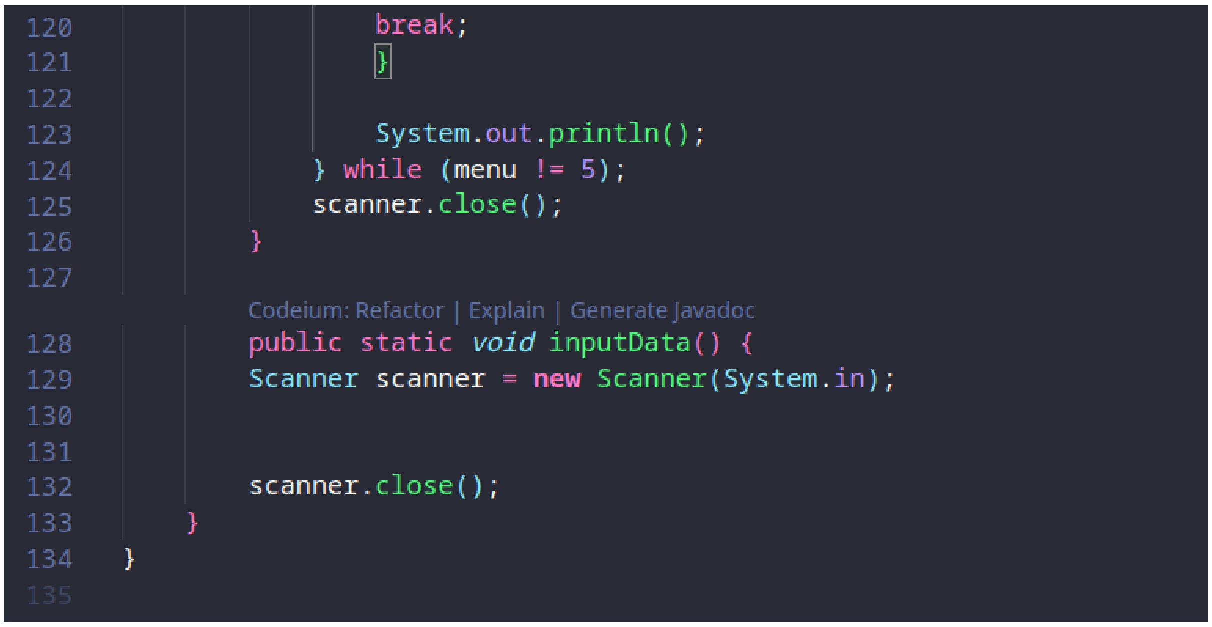
```
package assignment;

import java.util.*;

public class Main {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int menu;
        String studentId, courseStd;
        List<Grade> gradeList = new ArrayList<>();
        Grade grade = new Grade(student:null, course:null, grade:0);

        Student student1 = new Student(name:"Thalhah", studentId:"20001", telephone:"021xxx");
        Student student2 = new Student(name:"Zubair", studentId:"20002", telephone:"021xxx");
        Student student3 = new Student(name:"Abdur", studentId:"20003", telephone:"021xxx");
        Student student4 = new Student(name:"Sa'ad", studentId:"20004", telephone:"021xxx");
        Student student5 = new Student(name:"Sa'id", studentId:"20005", telephone:"021xxx");
        Student student6 = new Student(name:"Ubaidah", studentId:"20006", telephone:"021xxx");
        grade.addStudent(student1, student2, student3, student4, student5, student6);

        Course course1 = new Course(name:"Internet of Things", code:"00001", sks:3);
        Course course2 = new Course(name:"Data Structure and Algorithm", code:"00002", sks:2);
        Course course3 = new Course(name:"Algorithm and Programming", code:"00003",sks:2);
        Course course4 = new Course(name:"Data Structure and Algorithm Practicum", code:"00004", sks:3);
        Course course5 = new Course(name:"Algorithm and Programming Practicum", code:"00005", sks:3);
        grade.addCourse(course1, course2, course3, course4, course5);

        do {
            System.out.println("============================================");
            System.out.println("Semester Student Grade Data Processing System");
            System.out.println("============================================");
            System.out.println("1. Input Data");
            System.out.println("2. Print Data");
            System.out.println("3. Search Student's Grade");
            System.out.println("4. Sort Grades");
            System.out.println("5. Exit");
            System.out.println("============================================");
            System.out.print("Choose menu\t: ");
            menu = scanner.nextInt();

```



```java
            switch (menu) {
                case 1:
                System.out.println("Insert Data");
                System.out.print("Code\t: ");
                String code = scanner.next();
                System.out.print("grade\t: ");
                double gradeStd = scanner.nextDouble();
                System.out.println("List Student");
                System.out.println("=============================================");
                System.out.println("Student ID\tName\t\tTelephone");
                grade.printStudent();
                System.out.println("=============================================");
                System.out.print("Search Student by student id\t: ");
                studentId = scanner.next();
                Student student = grade.searchStudent(studentId);
                System.out.println("List Course");
                System.out.println("=============================================");
                System.out.println("Code\t\tCourse\t\t\t\t\t\tSks");
                grade.printCourse();
                System.out.println("=============================================");
                System.out.print("Search Course by code\t: ");
                courseStd = scanner.next();
                Course course = grade.searchCourse(courseStd);
                Grade grade1 = new Grade(student, course, gradeStd);
                gradeList.add(grade1);
                break;

                case 2:
                System.out.println("List Grade Student");
                System.out.println("=============================================");
                System.out.println("Student ID\tName\t\tCourse\t\t\t\tSKS\tGrade");
                for (int i = 0; i < gradeList.size(); i++) {
                    System.out.println(gradeList.get(i).toString());
                }
                break;

                case 3:
                System.out.println("List Grade Student");
                System.out.println("=============================================");
                System.out.println("Student ID\tName\t\tCourse\t\t\t\tSKS\tGrade");
```

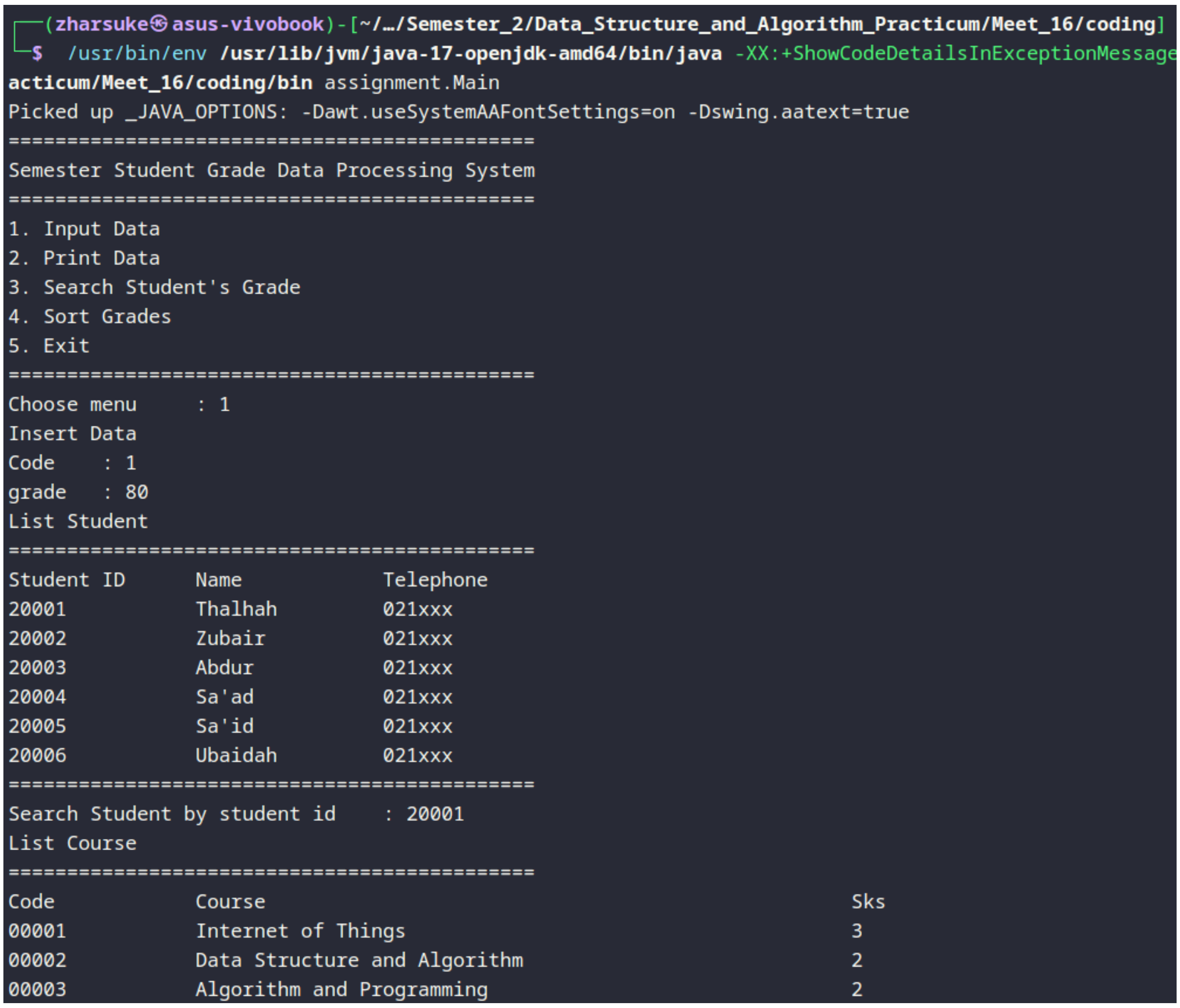


```java
                System.out.println("Student ID\tName\t\tCourse\t\t\t\tSKS\tGrade");
                for (int i = 0; i < gradeList.size(); i++) {
                    System.out.println(gradeList.get(i).toString());
                }
                System.out.println("============================================");
                System.out.print("Insert student data [Student ID]\t: ");
                studentId = scanner.next();
                int totalSks = 0;
                boolean isFound = true;
                System.out.println("Student ID\tName\t\tCourse\t\t\t\tSKS\tGrade");
                for (int i = 0; i < gradeList.size(); i++) {
                    if (gradeList.get(i).student.studentId.equals(studentId)) {
                        System.out.println(gradeList.get(i).toString());
                        totalSks += gradeList.get(i).course.sks;
                    }
                }
                if (!isFound) {
                    System.out.println("Data student with student id " + studentId + " not found");
                } else {
                    System.out.println("Total SKS " + totalSks + " already taken");
                }
                break;

                case 4:
                System.out.println("List Grade Student");
                System.out.println("============================================");
                System.out.println("Student ID\tName\t\tCourse\t\t\t\tSKS\tGrade");
                gradeList.sort(Comparator.comparing(g -> g.grade));
                for (int i = 0; i < gradeList.size(); i++) {
                    System.out.println(gradeList.get(i).toString());
                }
                break;

                case 5:
                System.out.println("Thank you. Exiting the program.");
                System.exit(0);
                break;

                default:
                System.out.println("Invalid menu choice.");
```

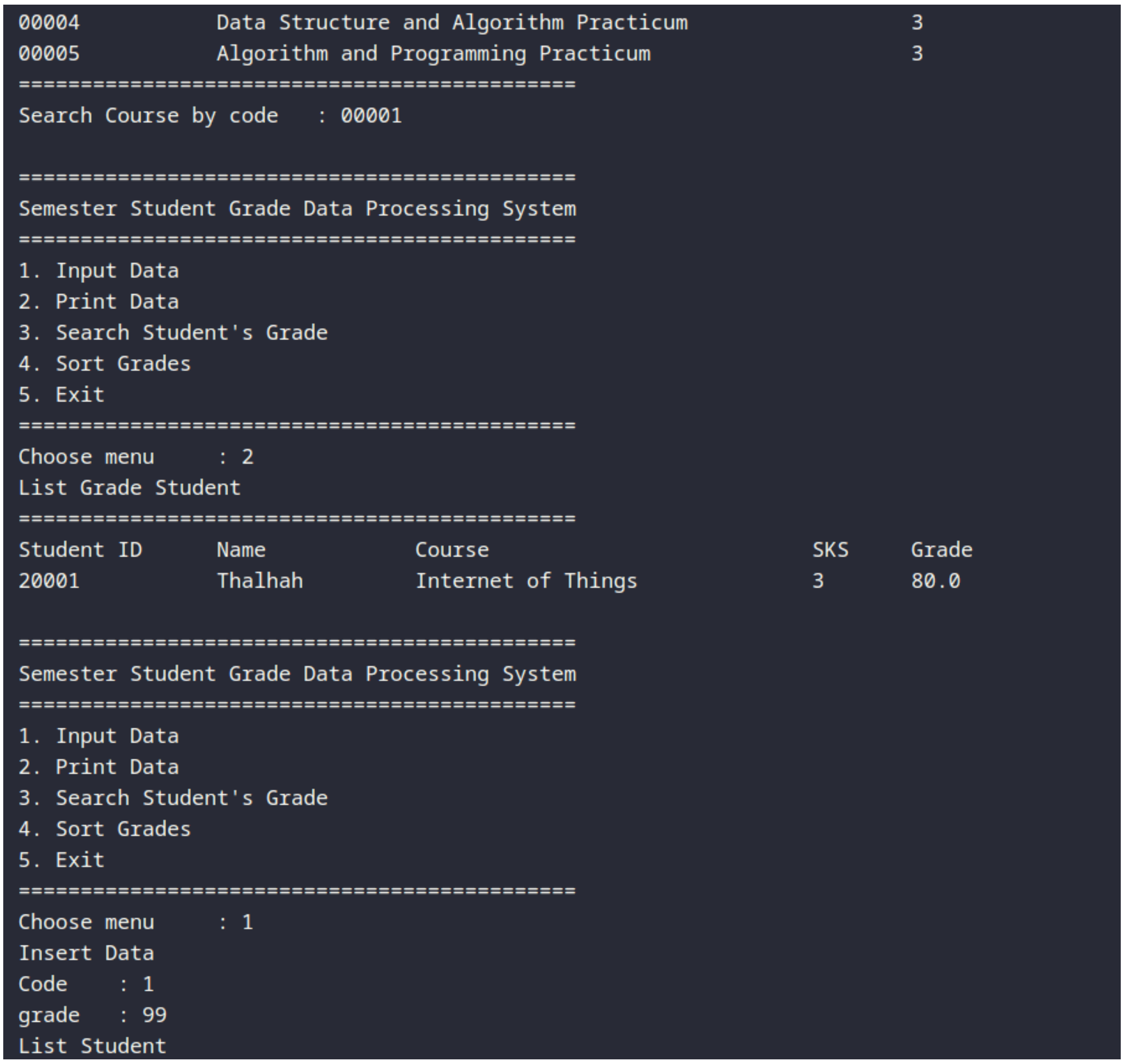


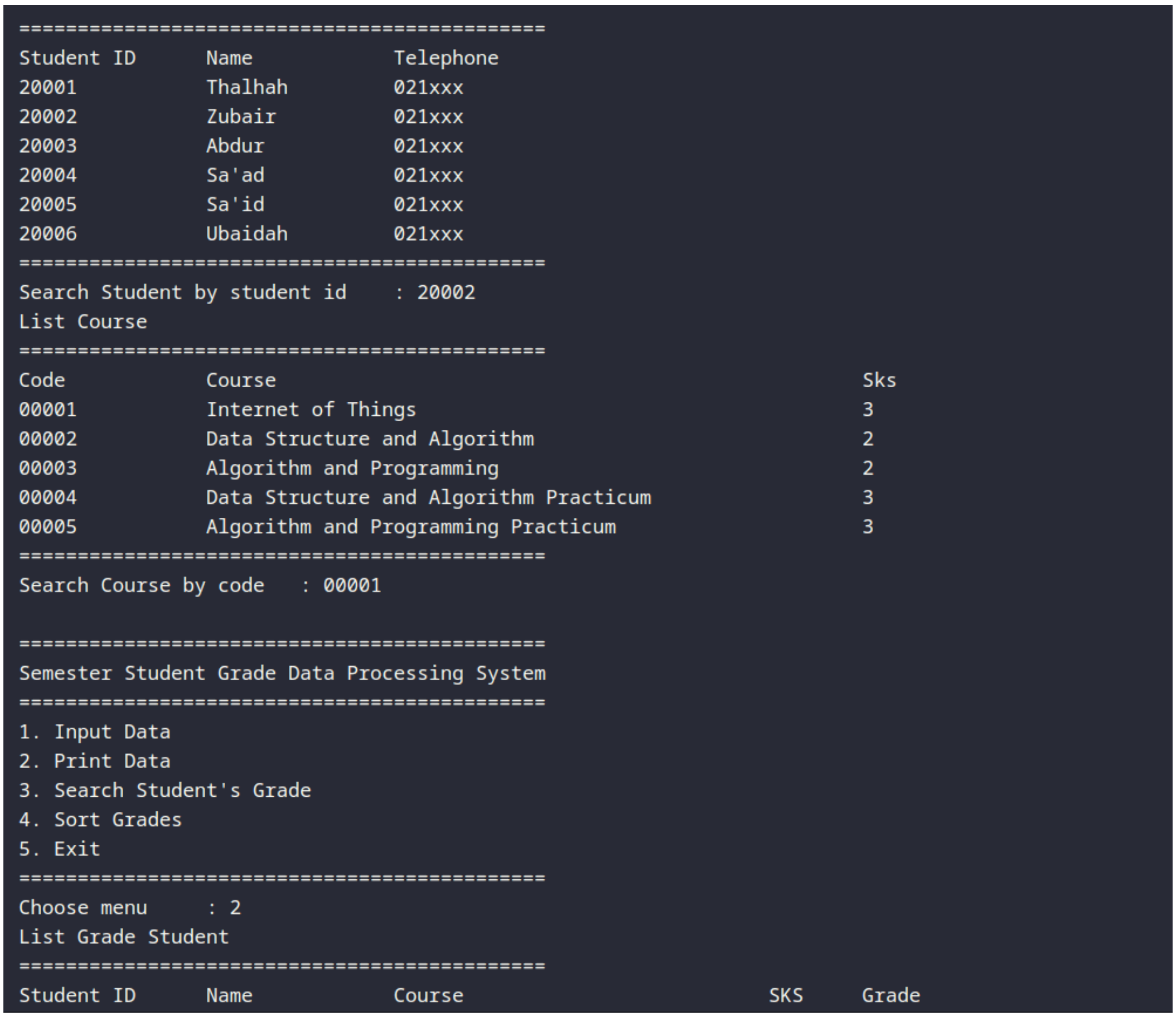
```java
                break;
                }

                System.out.println();
            } while (menu != 5);
            scanner.close();
        }

        public static void inputData() {
        Scanner scanner = new Scanner(System.in);


        scanner.close();
    }
}
```

- Result :

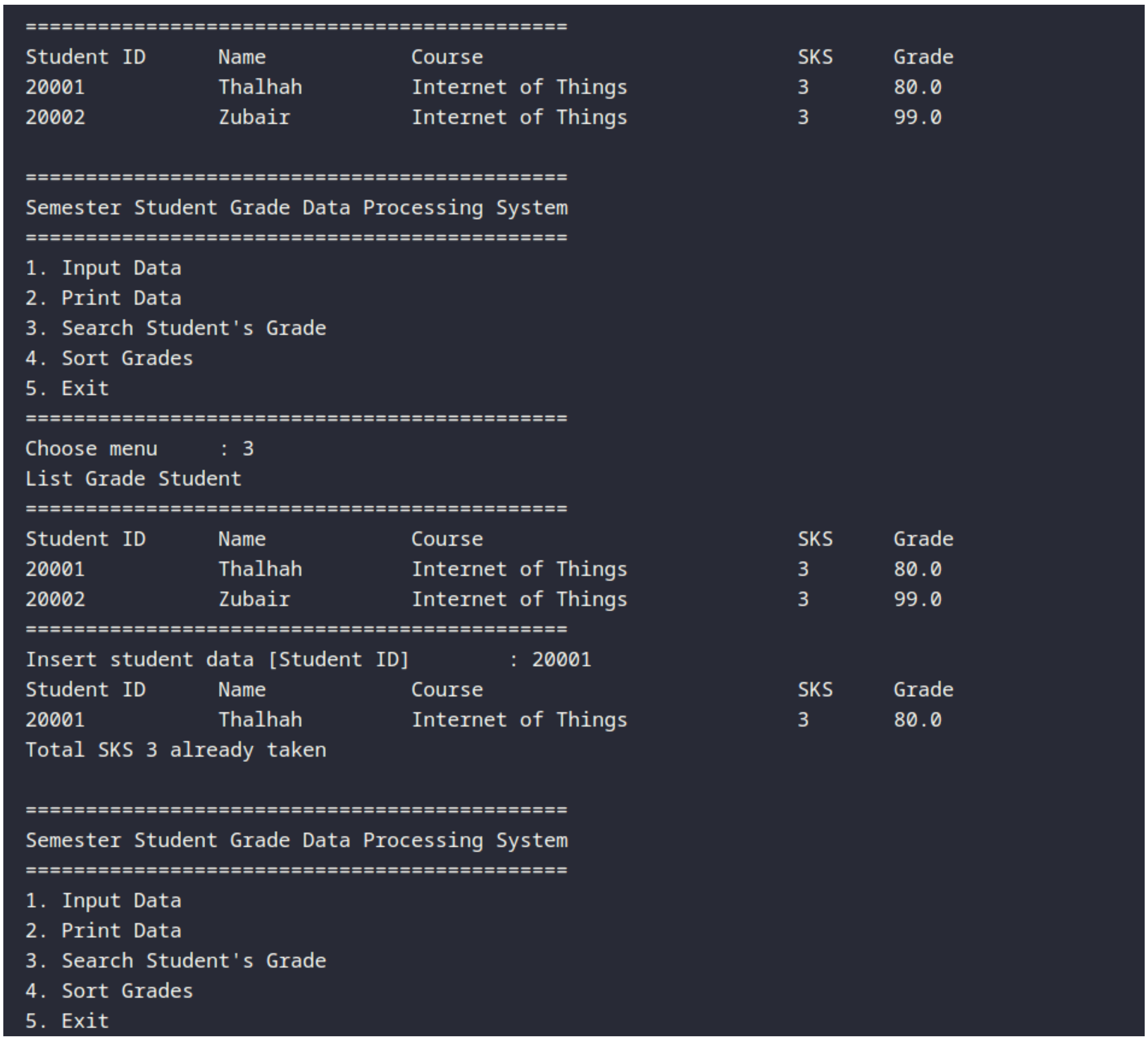


```
┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$  /usr/bin/env /usr/lib/jvm/java-17-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessage
acticum/Meet_16/coding/bin assignment.Main
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Exit
=============================================
Choose menu     : 1
Insert Data
Code    : 1
grade   : 80
List Student
=============================================
Student ID      Name            Telephone
20001           Thalhah         021xxx
20002           Zubair          021xxx
20003           Abdur           021xxx
20004           Sa'ad           021xxx
20005           Sa'id           021xxx
20006           Ubaidah         021xxx
=============================================
Search Student by student id    : 20001
List Course
=============================================
Code            Course                                                  Sks
00001           Internet of Things                                      3
00002           Data Structure and Algorithm                            2
00003           Algorithm and Programming                               2
```

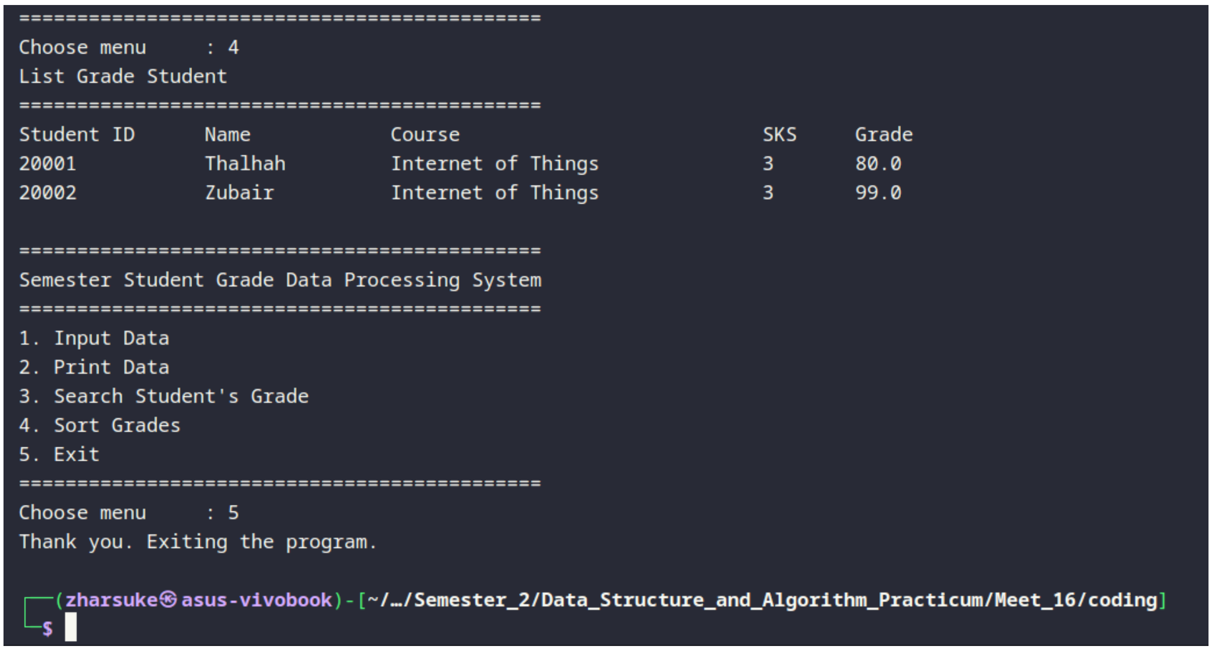

```
00004           Data Structure and Algorithm Practicum                  3
00005           Algorithm and Programming Practicum                     3
=============================================
Search Course by code   : 00001

=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Exit
=============================================
Choose menu     : 2
List Grade Student
=============================================
Student ID      Name            Course                          SKS     Grade
20001           Thalhah         Internet of Things              3       80.0

=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Exit
=============================================
Choose menu     : 1
Insert Data
Code    : 1
grade   : 99
List Student
```



```
=============================================
Student ID      Name            Telephone
20001           Thalhah         021xxx
20002           Zubair          021xxx
20003           Abdur           021xxx
20004           Sa'ad           021xxx
20005           Sa'id           021xxx
20006           Ubaidah         021xxx
=============================================
Search Student by student id    : 20002
List Course
=============================================
Code            Course                                          Sks
00001           Internet of Things                              3
00002           Data Structure and Algorithm                    2
00003           Algorithm and Programming                       2
00004           Data Structure and Algorithm Practicum          3
00005           Algorithm and Programming Practicum             3
=============================================
Search Course by code   : 00001

=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Exit
=============================================
Choose menu     : 2
List Grade Student
=============================================
Student ID      Name            Course                          SKS     Grade
```



```
=============================================
Student ID      Name            Course                          SKS     Grade
20001           Thalhah         Internet of Things              3       80.0
20002           Zubair          Internet of Things              3       99.0

=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Exit
=============================================
Choose menu     : 3
List Grade Student
=============================================
Student ID      Name            Course                          SKS     Grade
20001           Thalhah         Internet of Things              3       80.0
20002           Zubair          Internet of Things              3       99.0
=============================================
Insert student data [Student ID]        : 20001
Student ID      Name            Course                          SKS     Grade
20001           Thalhah         Internet of Things              3       80.0
Total SKS 3 already taken

=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Exit
```

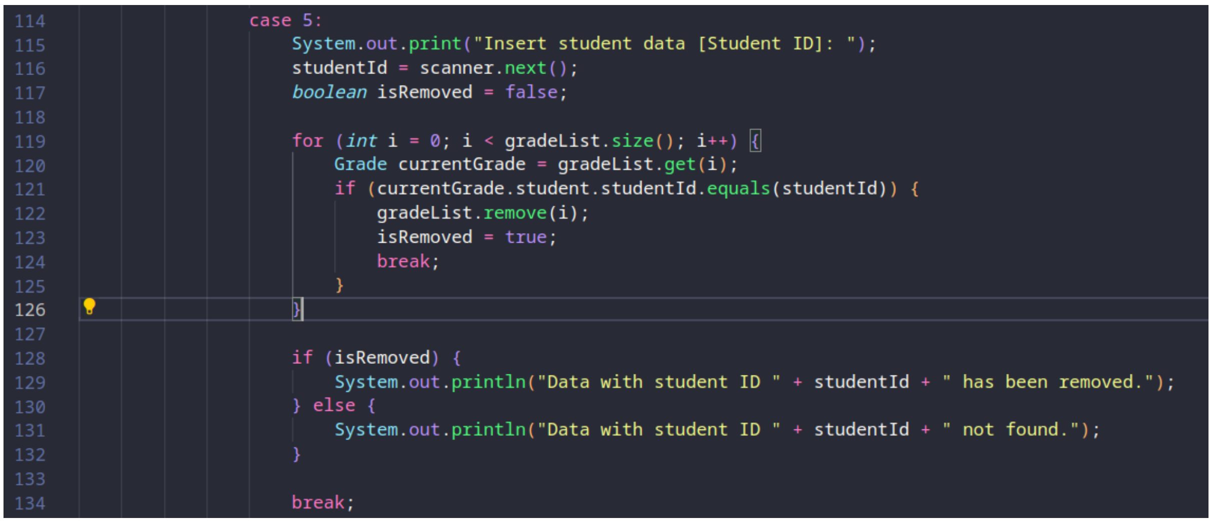


```
=============================================
Choose menu     : 4
List Grade Student
=============================================
Student ID      Name            Course                          SKS     Grade
20001           Thalhah         Internet of Things              3       80.0
20002           Zubair          Internet of Things              3       99.0

=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Exit
=============================================
Choose menu     : 5
Thank you. Exiting the program.

┌──(zharsuke㉿asus-vivobook)-[~/…/Semester_2/Data_Structure_and_Algorithm_Practicum/Meet_16/coding]
└─$ 
```

2. Add a procedure to delete student data through the implementation of Queue on collections

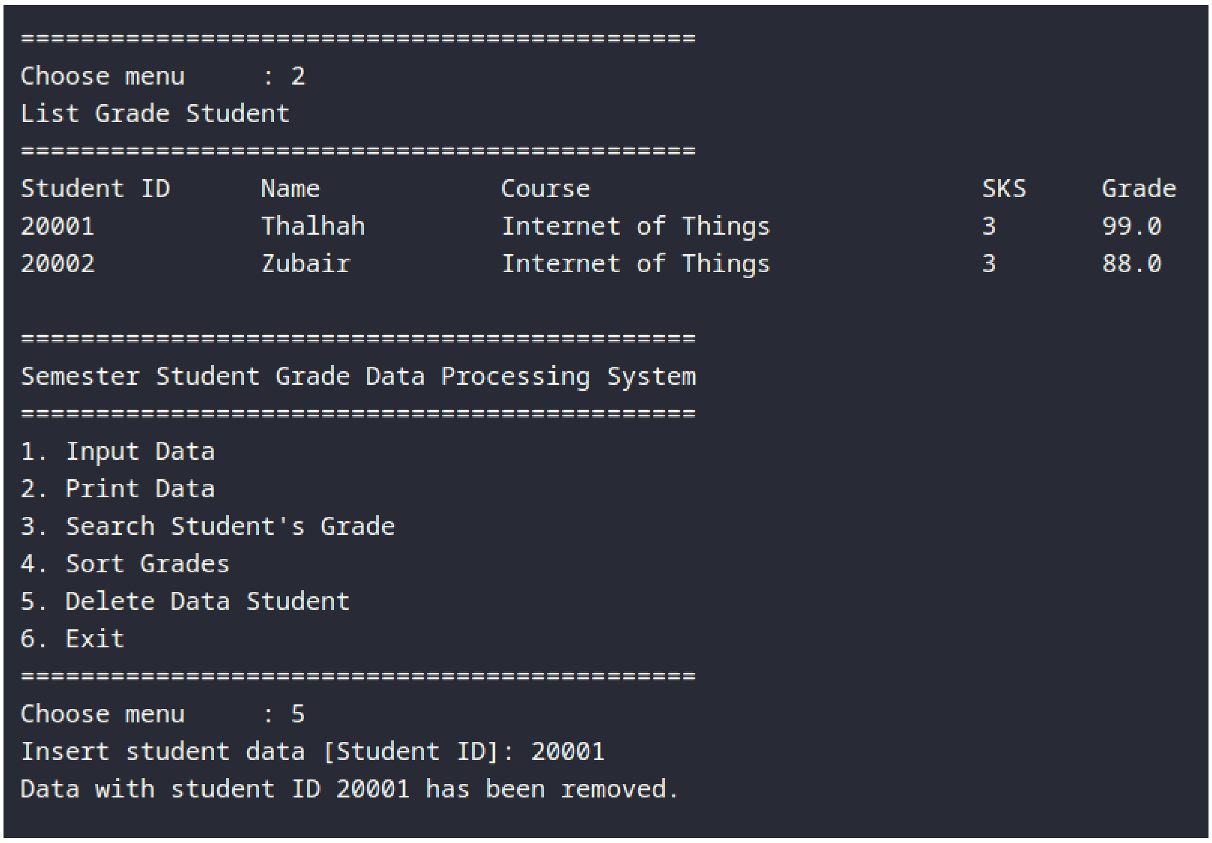
Task number 1!

code :

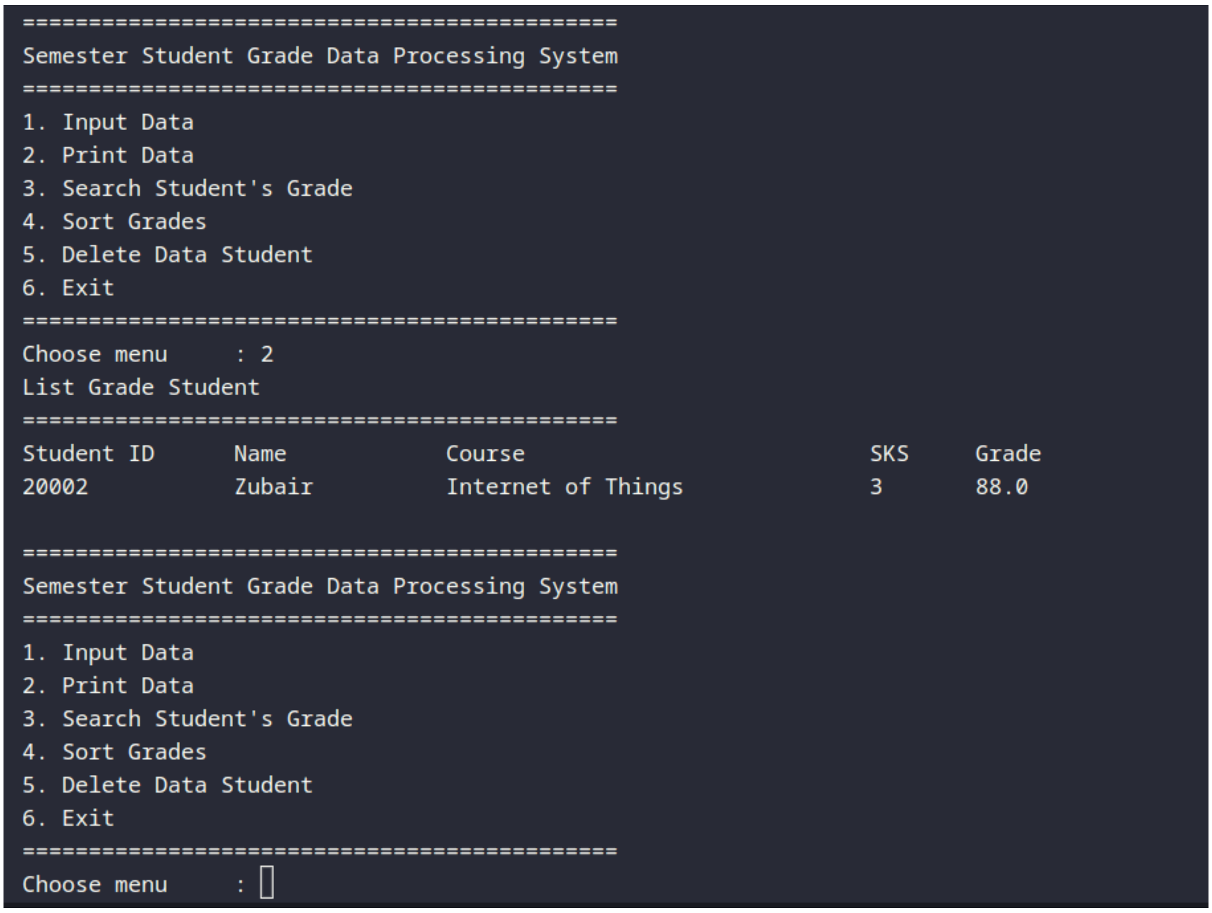


```java
                case 5:
                    System.out.print("Insert student data [Student ID]: ");
                    studentId = scanner.next();
                    boolean isRemoved = false;

                    for (int i = 0; i < gradeList.size(); i++) {
                        Grade currentGrade = gradeList.get(i);
                        if (currentGrade.student.studentId.equals(studentId)) {
                            gradeList.remove(i);
                            isRemoved = true;
                            break;
                        }
                    }

                    if (isRemoved) {
                        System.out.println("Data with student ID " + studentId + " has been removed.");
                    } else {
                        System.out.println("Data with student ID " + studentId + " not found.");
                    }

                    break;
```

Result :



```
=============================================
Choose menu      : 2
List Grade Student
=============================================
Student ID      Name            Course                          SKS     Grade
20001           Thalhah         Internet of Things              3       99.0
20002           Zubair          Internet of Things              3       88.0

=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Delete Data Student
6. Exit
=============================================
Choose menu      : 5
Insert student data [Student ID]: 20001
Data with student ID 20001 has been removed.
```



```
=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Delete Data Student
6. Exit
=============================================
Choose menu      : 2
List Grade Student
=============================================
Student ID      Name            Course                          SKS     Grade
20002           Zubair          Internet of Things              3       88.0

=============================================
Semester Student Grade Data Processing System
=============================================
1. Input Data
2. Print Data
3. Search Student's Grade
4. Sort Grades
5. Delete Data Student
6. Exit
=============================================
Choose menu      : 
```