



Name: Virak Rith

Student ID: P20230033

Course: OOP in java

Instructor: HOK Tin

Assignment: Lab04

Due Date: June 02, 2025 (11:59 AM)

Exercise 1

Lab04.1. Infinite Input of Student

Source Codes

```
import java.util.ArrayList;
import java.util.Scanner;

class Student {

    int id;
    String name;
    int age;

    Student(int id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }
}

public class ex1 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Student> studentArr = new ArrayList<>();
        int count = 0;
        String answer;
        do {
            System.out.println("Student #" + (count + 1));
            System.out.print("ID: ");
            int id = Integer.parseInt(scanner.nextLine());
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Age: ");
            int age = Integer.parseInt(scanner.nextLine());
```

```

        studentArr.add(new Student(id, name, age));
        count++;
        System.out.print("Do you want to add more (y/n): ");
        answer = scanner.nextLine();
    } while (answer.equalsIgnoreCase("y"));
    for (int i = 0; i < 48; i++) {
        System.out.print("=");
    }
    System.out.printf("\n| %-5s | %-10s | %-15s | %-5s |\n", "No", "ID",
        "Name", "Age");
    for (int i = 0; i < 48; i++) {
        System.out.print("=");
    }
    System.out.println();
    for (int i = 0; i < studentArr.size(); i++) {
        Student student = studentArr.get(i);
        System.out.printf("| %-5d | %-10d | %-15s | %-5d |\n", (i + 1),
            student.id, student.name, student.age);
    }
    for (int i = 0; i < 48; i++) {
        System.out.print("=");
    }
    scanner.close();
}
}

```

Output

```

Student #1
ID: 101
Name: Dara
Age: 20
Do you want to add more (y/n): y
Student #2
ID: 102
Name: Sopheak
Age: 21
Do you want to add more (y/n): y
Student #3
ID: 103
Name: Lina
Age: 19
Do you want to add more (y/n): n
=====
| No    | ID      | Name      | Age  |
=====
| 1     | 101     | Dara      | 20   |
| 2     | 102     | Sopheak   | 21   |
| 3     | 103     | Lina      | 19   |
=====

```

Exercise 2

Lab04.2. Student management without using ArrayList

Source Codes

```
import java.util.Scanner;

class Student {

    int id;
    String name;
    int age;

    Student(int id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }
}

public class ex2 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Student[] students = new Student[100];
        int studentCount = 0;
        boolean processing = true;
        while (processing) {
            System.out.println("==== Menu ====");
            System.out.println("1. View all students");
            System.out.println("2. Add a new student");
            System.out.println("3. Delete a student");
            System.out.println("4. Quit");
            System.out.print("Choose an option: ");
            int option = Integer.parseInt(scanner.nextLine());
            switch (option) {
                case 1:
                    for (int i = 0; i < 48; i++) {
                        System.out.print("=");
                    }
                    System.out.printf("\n| %-5s | %-10s | %-15s | %-5s |\n", "No",
                        "ID", "Name", "Age");
                    for (int i = 0; i < 48; i++) {
                        System.out.print("=");
                    }
                    System.out.println();
                    for (int i = 0; i < studentCount; i++) {
                        Student student = students[i];
```

```

        System.out.printf("| %-5d | %-10d | %-15s | %-5d |\n", (i
            + 1), student.id, student.name,
            student.age);
    }
    for (int i = 0; i < 48; i++) {
        System.out.print("=");
    }
    System.out.println();
    break;
case 2:
    if (studentCount >= students.length) {
        System.out.println("Cannot add more students. List is
full.");
        break;
    }
    System.out.println("==== Add a new student =====");
    System.out.print("ID: ");
    int id = Integer.parseInt(scanner.nextLine());
    System.out.print("Name: ");
    String name = scanner.nextLine();
    System.out.print("Age: ");
    int age = Integer.parseInt(scanner.nextLine());
    students[studentCount++] = new Student(id, name, age);
    System.out.println("Student added successfully.");
    break;
case 3:
    if (studentCount == 0) {
        System.out.println("There are no students in the list
yet!");
        break;
    } else {
        System.out.println("==== Delete a student =====");
        while (true) {
            System.out.print("Input student ID: ");
            int deleteId = Integer.parseInt(scanner.nextLine());
            boolean found = false;
            for (int i = 0; i < studentCount; i++) {
                if (students[i].id == deleteId) {
                    System.out.println(
                        "The following student has been
deleted:");
                    for (int j = 0; j < 48; j++) {
                        System.out.print("=");
                    }
                    System.out.println();
                    System.out.printf("| %-5d | %-10d | %-15s |
%-5d|\n", (i + 1), students[i].id, students[i].name, students[i].age
                    );
                    for (int j = 0; j < 48; j++) {
                        System.out.print("=");
                    }
                    System.out.println();
                    for (int j = i; j < studentCount - 1; j++) {
                        students[j] = students[j + 1];

```

```

        }
        students[studentCount--] = null;
        found = true;
        break;
    }
}
if (!found) {
    System.out.println("Student is not found. Try
again.");
} else {
    break;
}
}
break;
}
case 4:
    processing = false;
    System.out.println("Thank you for using our program<<");
    break;
default:
    System.out.println("Invalid option. Please try again...");
}
}
scanner.close();
}
}

```

Output

```

==== Menu ====
1. View all students
2. Add a new student
3. Delete a student
4. Quit
Choose an option: 1
=====
| No    | ID      | Name          | Age  |
|-----|-----|-----|-----|
=====
==== Menu ====
1. View all students
2. Add a new student
3. Delete a student
4. Quit
Choose an option: 2
==== Add a new student ====
ID: 101
Name: Alice
Age: 20
Student added successfully.

```

```
==== Menu ====
1. View all students
2. Add a new student
3. Delete a student
4. Quit
Choose an option: 2
==== Add a new student ====
ID: 102
Name: Bob
Age: 22
Student added successfully.
==== Menu ====
1. View all students
2. Add a new student
3. Delete a student
4. Quit
Choose an option: 1
=====
| No      | ID      | Name      | Age      |
=====
| 1       | 101     | Alice     | 20       |
| 2       | 102     | Bob       | 22       |
=====
==== Menu ====
1. View all students
2. Add a new student
3. Delete a student
4. Quit
Choose an option: 3
==== Delete a student ====
Input student ID: 101
The following student has been deleted:
=====
| 1       | 101     | Alice     | 20       |
=====
==== Menu ====
1. View all students
2. Add a new student
3. Delete a student
4. Quit
Choose an option: 1
=====
| No      | ID      | Name      | Age      |
=====
| 1       | 102     | Bob       | 22       |
=====
==== Menu ====
1. View all students
2. Add a new student
3. Delete a student
4. Quit
Choose an option: 3
==== Delete a student ====
Input student ID: 999
```

```

Student is not found. Try again.
Input student ID: 102
The following student has been deleted:
=====
| 1      | 102      | Bob      | 22      |
=====
==== Menu ====
1. View all students
2. Add a new student
3. Delete a student
4. Quit
Choose an option: 1
=====
| No      | ID      | Name      | Age      |
=====
=====
==== Menu ====
1. View all students
2. Add a new student
3. Delete a student
4. Quit
Choose an option: 4
Thank you for using our program><

```

Exercise 3

Lab04.3. Book store management without using ArrayList

Source Codes

```

import java.util.Scanner;

class Book {

    int isbn;
    String title;
    double price;
    String author;

    Book(int isbn, String title, double price, String author) {
        this.isbn = isbn;
        this.title = title;
        this.price = price;
        this.author = author;
    }

}

```

```
public class ex3 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        Book[] books = new Book[100];
        int bookCount = 0;
        int choice;

        do {

            System.out.println("=== Menu ===");
            System.out.println("1. View all books");
            System.out.println("2. Add a new book");
            System.out.println("3. Update a book");
            System.out.println("4. Quit");
            System.out.print("Choose an option: ");
            choice = scanner.nextInt();

            switch (choice) {

                case 1:

                    if (bookCount == 0) {
                        System.out.println("No books available yet!");
                    } else {

                        for (int i = 0; i < 57; i++) {
                            System.out.print("=");
                        }
                        System.out.println();

                        System.out.printf("| %-5s | %-18s | %-6s | %-15s |\n",
"ISBN", "Title", "Price", "Author");

                        for (int i = 0; i < 57; i++) {
                            System.out.print("=");
                        }
                        System.out.println();

                        for (int i = 0; i < bookCount; i++) {
                            Book b = books[i];
                            System.out.printf("| %-5d | %-18s | %6.2f | %-15s
|\n", b.isbn, b.title, b.price, b.author);
                        }

                        for (int i = 0; i < 57; i++) {
                            System.out.print("=");
                        }
                        System.out.println();
                    }

                    break;
            }
        }
    }
}
```



```

        case 2:

            if (bookCount >= books.length) {
                System.out.println("Cannot add more book. List is full.");
                break;
            }

            System.out.println("=== Add a new Book ===");
            scanner.nextLine();
            System.out.print("ISBN: ");
            int isbn = Integer.parseInt(scanner.nextLine());
            System.out.print("Title: ");
            String title = scanner.nextLine();
            System.out.print("Price: ");
            double price = Double.parseDouble(scanner.nextLine());
            System.out.print("Author: ");
            String author = scanner.nextLine();

            books[bookCount++] = new Book(isbn, title, price, author);

            System.out.println("The book has been added to the list
successfully!");
            break;

        case 3:

            if (bookCount == 0) {
                System.out.println("No books available to update!");
                break;
            } else {

                scanner.nextLine();
                System.out.println("=== Update a book ===");
                boolean found = false;

                while (true) {
                    System.out.print("Input ISBN: ");
                    int isbnAdjust = Integer.parseInt(scanner.nextLine());

                    for (int i = 0; i < bookCount; i++) {
                        if (books[i].isbn == isbnAdjust) {
                            System.out.println("Please update the
followoing:");

                            System.out.print("ISBN: ");
                            int newIsbn =
Integer.parseInt(scanner.nextLine());
                            System.out.print("Title: ");
                            String newTitle = scanner.nextLine();
                            System.out.print("Price: ");
                            double newPrice =
Double.parseDouble(scanner.nextLine());
                            System.out.print("Author: ");
                            String newAuthor = scanner.nextLine();

```

```

                                books[i] = new Book(newIsbn, newTitle,
newPrice, newAuthor);
                                System.out.println("Book updated
successfully!");
                                found = true;
                                break;
                            }
                        }
                    if (!found) {
                        System.out.println("Book is not found. Try
again...");
                    } else {
                        break;
                    }
                }
            }
            break;
        case 4:
            System.out.println("Thank you for using our program><");
            break;
        default:
            System.out.println("Invaild input, please try again...");
            break;
    }
} while (choice != 4);
scanner.close();
}
}

```

Output

```

=== Menu ===
1. View all books
2. Add a new book
3. Update a book
4. Quit
Choose an option: 1
No books available yet!
=== Menu ===
1. View all books
2. Add a new book
3. Update a book

```

```
4. Quit
Choose an option: 2
=== Add a new Book ===
ISBN: 12345
Title: Java Basics
Price: 29.99
Author: John Doe
The book has been added to the list successfully!
=== Menu ===
1. View all books
2. Add a new book
3. Update a book
4. Quit
Choose an option: 2
=== Add a new Book ===
ISBN: 67890
Title: Data Structures
Price: 39.50
Author: Jane Smith
The book has been added to the list successfully!
=== Menu ===
1. View all books
2. Add a new book
3. Update a book
4. Quit
Choose an option: 1
=====
| ISBN | Title                | Price | Author                |
=====
| 12345 | Java Basics         | 29.99 | John Doe              |
| 67890 | Data Structures      | 39.50 | Jane Smith            |
=====
=== Menu ===
1. View all books
2. Add a new book
3. Update a book
4. Quit
Choose an option: 3
=== Update a book ===
Input ISBN: 12345
Please update the follwoing:
ISBN: 12345
Title: Advanced Java
Price: 35.00
Author: John Doe
Book updated successfully!
=== Menu ===
1. View all books
2. Add a new book
3. Update a book
4. Quit
Choose an option: 1
=====
| ISBN | Title                | Price | Author                |
```

```
=====
| 12345 | Advanced Java      | 35.00 | John Doe      |
| 67890 | Data Structures      | 39.50 | Jane Smith    |
=====

=== Menu ===
1. View all books
2. Add a new book
3. Update a book
4. Quit
Choose an option: 4
Thank you for using our program><
```

Exercise 4

Lab04.4. Video Search Engine without using ArrayList

Source Codes

```
import java.util.Scanner;

class Video {

    String title;
    String uploader;
    double length;
    String type;

    Video(String title, String uploader, double length, String type) {
        this.title = title;
        this.uploader = uploader;
        this.length = length;
        this.type = type;
    }
}

public class ex4 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        Video[] videos = new Video[100];
        int videoCount = 0;
        videos[videoCount++] = new Video("Hello word", "Rith", 123, "heheh");
        videos[videoCount++] = new Video("Hello", "Rithy", 123, "heheh");
        int choice;

        do {
```

```
System.out.println("=== Menu ===");
System.out.println("1. Search");
System.out.println("2. View all videos");
System.out.println("3. Add a new video");
System.out.println("4. Quit");
System.out.print("Choose an option: ");
choice = Integer.parseInt(scanner.nextLine());

switch (choice) {

    case 1 -> {
        if (videoCount == 0) {
            System.out.println("No videos available yet!");
            break;
        }

        System.out.println("==== Video Search ====");
        System.out.print("Input Uploader: ");
        String searchUploader = scanner.nextLine().toLowerCase();

        int foundCount = 0;
        for (int i = 0; i < videoCount; i++) {
            if
(videos[i].uploader.toLowerCase().contains(searchUploader)) {
                if (foundCount == 0) {
                    System.out.println("Videos found:");
                }
                System.out.printf("%d. %s by %s\n", (foundCount++ +
1), videos[i].title, videos[i].uploader);
            }
        }

        if (foundCount == 0) {
            System.out.println("0 video found. Try again...");
        }
    }

    case 2 -> {
        if (videoCount == 0) {
            System.out.println("No videos available yet!");
        } else {
            for (int i = 0; i < 75; i++) {
                System.out.print("=");
            }
            System.out.println();

            System.out.printf("| %-3s | %-20s | %-15s | %-6s | %-15s
|\n", "No", "Title", "Uploader", "Length",
                "Type");

            for (int i = 0; i < 75; i++) {
                System.out.print("=");
            }
            System.out.println();
        }
    }
}
```

```

        for (int i = 0; i < videoCount; i++) {
            Video v = videos[i];
            System.out.printf("| %-3d | %-20s | %-15s | %6.2f |
%-15s |\n", (i + 1), v.title, v.uploader,
                        v.length, v.type);
        }

        for (int i = 0; i < 75; i++) {
            System.out.print("=");
        }
        System.out.println();
    }

    case 3 -> {
        if (videoCount >= videos.length) {
            System.out.println("Cannot add more videos. List is
full.");

            break;
        }

        System.out.println("==== Add a new video ====");
        System.out.print("Title: ");
        String title = scanner.nextLine();
        System.out.print("Uploader: ");
        String uploader = scanner.nextLine();
        System.out.print("Length (in minutes): ");
        double length = Double.parseDouble(scanner.nextLine());
        System.out.print("Type: ");
        String type = scanner.nextLine();

        videos[videoCount++] = new Video(title, uploader, length,
type);

        System.out.println("The video has been added to the list
successfully!");
    }

    case 4 ->
        System.out.println("Thank you for using our Video Search
Engine! Goodbye!");

    default ->
        System.out.println("Invalid input. Please try again...");
    }

    } while (choice != 4);

    scanner.close();
}

```

Output

```
=== Menu ===
1. Search
2. View all videos
3. Add a new video
4. Quit
Choose an option: 2
=====
| No  | Title                | Uploader          | Length | Type           |
=====
| 1   | Hello word           | Rith              | 123.00 | heheh          |
| 2   | Hello                | Rithy             | 123.00 | heheh          |
=====
=== Menu ===
1. Search
2. View all videos
3. Add a new video
4. Quit
Choose an option: 1
===== Video Search =====
Input Uploader: rith
Videos found:
1. Hello word by Rith
2. Hello by Rithy
=== Menu ===
1. Search
2. View all videos
3. Add a new video
4. Quit
Choose an option: 1
===== Video Search =====
Input Uploader: y
Videos found:
1. Hello by Rithy
=== Menu ===
1. Search
2. View all videos
3. Add a new video
4. Quit
Choose an option: 3
===== Add a new video =====
Title: Learning Java
Uploader: John
Length (in minutes): 45.5
Type: tutorial
The video has been added to the list successfully!
=== Menu ===
1. Search
2. View all videos
3. Add a new video
4. Quit
```

Choose an option: 2

=====					
No	Title	Uploader	Length	Type	
=====					
1	Hello word	Rith	123.00	heheh	
2	Hello	Rithy	123.00	heheh	
3	Learning Java	John	45.50	tutorial	
=====					

=== Menu ===

1. Search
2. View all videos
3. Add a new video
4. Quit

Choose an option: 4

Thank you for using our Video Search Engine! Goodbye!

Exercise 5

Lab04.5. Music Store

Source Codes

```
import java.util.Scanner;

class Color {

    public static final String RESET = "\u001B[0m";
    public static final String RED = "\u001B[31m";
    public static final String GREEN = "\u001B[32m";
}

class Album {

    public String title;
    public String genre;
    public Song[] songs;
    public int songCount;

    public Album(String title, String genre) {
        this.title = title;
        this.genre = genre;
        this.songs = new Song[100];
        this.songCount = 0;
    }

    public void addSong(Song song) {
        if (songCount < songs.length) {
```



```

        songs[songCount++] = song;
    } else {
        System.out.println("Cannot add more songs.");
    }
}

public void displaySongs() {
    if (songCount == 0) {
        System.out.println("None of song");
    } else {
        for (int i = 0; i < songCount; i++) {
            System.out.println("| " + (i + 1) + songs[i]);
        }
    }
}
}

class Song {

    public String title;
    public String singer;
    public int length;
    public double price;

    public Song(String title, String singer, int length, double price) {
        this.title = title;
        this.singer = singer;
        this.length = length;
        this.price = price;
    }

    @Override
    public String toString() {
        return String.format(" | %-15s | %-15s | %-5s | %-8.2f$ |", title, singer,
length + "mins", price);
    }
}

public class ex5 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Album[] albums = new Album[100];
        int albumCount = 0;

        OUTER:
        while (true) {
            System.out.println("\n===== Menu =====");
            System.out.println("1. View a music store");
            System.out.println("2. Add a song");
            System.out.println("3. Create an album");
            System.out.println("4. Quit");
            System.out.print("Choose an option: ");
            int option = scanner.nextInt();

```

```

        scanner.nextLine();
        switch (option) {
            case 1 -> {
                System.out.println("===== Music Store =====");
                if (albumCount == 0) {
                    System.out.println(Color.RED + "No albums yet." +
Color.RESET);
                } else {
                    for (int i = 0; i < albumCount; i++) {
                        System.out.println("Album: " + albums[i].title);
                        albums[i].displaySongs();
                    }
                }
            }
            case 2 -> {
                if (albumCount == 0) {
                    System.out.println(Color.RED + "No albums available.
Please create one first." + Color.RESET);
                    continue;
                }
                System.out.println("===== Add a new song =====");
                System.out.println("Select following album:");
                for (int i = 0; i < albumCount; i++) {
                    System.out.println((i + 1) + ". " + albums[i].title);
                }
                System.out.print("Choose an option: ");
                int albumChoice = scanner.nextInt();
                scanner.nextLine();
                if (albumChoice < 1 || albumChoice > albumCount) {
                    System.out.println(Color.RED + "Invalid album choice." +
Color.RESET);
                    continue;
                }
                System.out.print("Song title: ");
                String songTitle = scanner.nextLine();
                System.out.print("Signer: ");
                String singer = scanner.nextLine();
                System.out.print("Length: ");
                int length =
Integer.parseInt(scanner.nextLine().replace("mins", "").trim());
                System.out.print("Price: ");
                double price = scanner.nextDouble();
                scanner.nextLine();
                Song song = new Song(songTitle, singer, length, price);
                albums[albumChoice - 1].addSong(song);
                System.out.println(Color.GREEN + "A new song added to the
album" + Color.RESET);
            }
            case 3 -> {
                System.out.println("===== Create new album =====");
                System.out.print("Album title: ");
                String albumTitle = scanner.nextLine();
                System.out.print("Genre: ");
                String genre = scanner.nextLine();

```

```

        albums[albumCount++] = new Album(albumTitle, genre);
        System.out.println(Color.GREEN + "Album Created Successfully!"
+ Color.RESET);
    }
    case 4 -> {
        System.out.println("Thank you. Goodbye!");
        break OUTER;
    }
    default -> {
        System.out.println(Color.RED + "Invalid option." +
Color.RESET);
        System.out.println();
    }
}
}
scanner.close();
}
}

```

Output

```

===== Menu =====
1. View a music store
2. Add a song
3. Create an album
4. Quit
Choose an option: 1
===== Music Store =====
□[31mNo albums yet.□[0m

===== Menu =====
1. View a music store
2. Add a song
3. Create an album
4. Quit
Choose an option: 3
===== Create new album =====
Album title: My Album
Genre: Pop
□[32mAlbum Created Successfully!□[0m

===== Menu =====
1. View a music store
2. Add a song
3. Create an album
4. Quit
Choose an option: 2
===== Add a new song =====
Select following album:

```

```

1. My Album
Choose an option: 1
Song title: Song One
Signer: Singer A
Length (mins): 4
Price: 1.99
☐[32mA new song added to the album☐[0m

```

```

===== Menu =====
1. View a music store
2. Add a song
3. Create an album
4. Quit
Choose an option: 1
===== Music Store =====
Album: My Album
| 1 | Song One          | Singer A          | 4mins | 1.99$  |

```

```

===== Menu =====
1. View a music store
2. Add a song
3. Create an album
4. Quit
Choose an option: 4
Thank you. Goodbye!

```

Exercise 6

Lab04.6. Storing a list of Student without using any Array

Source Codes

```

class StudentList {

    private class Element {

        String name;
        Element next;

        Element(String name) {
            this.name = name;
            this.next = null;
        }
    }

    private Element head = null;
}

```

```
public void add(String name) {
    Element newElement = new Element(name);
    if (head == null) {
        head = newElement;
    } else {
        Element current = head;
        while (current.next != null) {
            current = current.next;
        }
        current.next = newElement;
    }
    System.out.println("Added: " + name);
}

public void removeLast() {
    if (head == null) {
        System.out.println("List is empty. Cannot remove.");
        return;
    }

    if (head.next == null) {
        System.out.println("Removed: " + head.name);
        head = null;
        return;
    }

    Element previous = head;
    Element current = head.next;

    while (current.next != null) {
        previous = current;
        current = current.next;
    }

    System.out.println("Removed: " + current.name);
    previous.next = null;
}

public void remove(int index) {
    if (head == null) {
        System.out.println("List is empty. Cannot remove at index " + index);
        return;
    }

    if (index == 0) {
        System.out.println("Removed at index 0: " + head.name);
        head = head.next;
        return;
    }

    Element previous = head;
    Element current = head.next;
    int i = 1;
```

```
        while (current != null && i < index) {
            previous = current;
            current = current.next;
            i++;
        }

        if (current == null) {
            System.out.println("Index " + index + " is out of range.");
        } else {
            System.out.println("Removed at index " + index + ": " + current.name);
            previous.next = current.next;
        }
    }

    public void clear() {
        head = null;
        System.out.println("All elements cleared.");
    }

    public void display() {
        if (head == null) {
            System.out.println("Array is empty");
            return;
        }

        Element current = head;
        System.out.println("Student List:");
        while (current != null) {
            System.out.println("- " + current.name);
            current = current.next;
        }
    }
}

public class ex6 {

    public static void main(String[] args) {
        StudentList list = new StudentList();

        System.out.println("==> Add");
        list.add("Makara");
        list.add("Kompheak");
        list.display();
        System.out.println("\n==> Remove at last");
        list.removeLast();
        list.add("Minea");
        list.add("Mehsa");
        list.display();
        System.out.println("\n==> Remove at larger index");
        list.remove(10);
        list.display();
        System.out.println("==> Clear all element");
        list.clear();
        list.display();
    }
}
```

```
}  
}
```

Output

```
==> Add  
Added: Makara  
Added: Kompheak  
Student List:  
- Makara  
- Kompheak  
  
==> Remove at last  
Removed: Kompheak  
Added: Minea  
Added: Mehsa  
Student List:  
- Makara  
- Minea  
- Mehsa  
  
==> Remove at larger index  
Index 10 is out of range.  
Student List:  
- Makara  
- Minea  
- Mehsa  
==> Clear all element  
All elements cleared.  
Array is empty
```

Exercise 7

Lab04.7. Inheritance in Java

Part1

Source Codes

```
class Point {  
  
    int x, y;  
  
    Point(int x, int y) {
```

```
        this.x = x;
        this.y = y;
    }

    void displayPoint() {
        System.out.println("Point: (" + x + ", " + y + ")");
    }
}

class Line extends Point {

    Point end;

    Line(int x1, int y1, int x2, int y2) {
        super(x1, y1);
        this.end = new Point(x2, y2);
    }

    void displayLine() {
        displayPoint();
        System.out.println("Line: Start(" + x + ", " + y + "), End(" + end.x + ", " + end.y + ")");
    }
}

class Triangle extends Line {

    Point thirdPoint;

    Triangle(int x1, int y1, int x2, int y2, int x3, int y3) {
        super(x1, y1, x2, y2);
        this.thirdPoint = new Point(x3, y3);
    }

    void displayTriangle() {
        displayLine();
        System.out.println("Third Point: (" + thirdPoint.x + ", " + thirdPoint.y + ")");
    }
}

class Rectangle extends Triangle {

    Point fourthPoint;

    Rectangle(int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4) {
        super(x1, y1, x2, y2, x3, y3);
        this.fourthPoint = new Point(x4, y4);
    }

    public void displayRectangle() {
        displayTriangle();
        System.out.println("Fourth Point: (" + fourthPoint.x + ", " + fourthPoint.y + ")");
    }
}
```



```
    }  
}  
  
public class ex7_1 {  
  
    public static void main(String[] args) {  
        Rectangle r = new Rectangle(0, 0, 4, 0, 4, 3, 0, 3);  
        r.displayRectangle();  
    }  
}
```

Output

```
Point: (0, 0)  
Line: Start(0, 0), End(4, 0)  
Third Point: (4, 3)  
Fourth Point: (0, 3)
```

Part2

Source Codes

```
class ITStudent {  
  
    String name;  
    int id;  
  
    ITStudent(String name, int id) {  
        this.name = name;  
        this.id = id;  
    }  
  
    void showInfo() {  
        System.out.println("IT Student Name: " + name + ", ID: " + id);  
    }  
}  
  
class ProgrammingStudent extends ITStudent {  
  
    String language;  
  
    ProgrammingStudent(String name, int id, String language) {  
        super(name, id);  
        this.language = language;  
    }  
  
    void displayProgramming() {
```

```
        showInfo();
        System.out.println("Specialization: Programming in " + language);
    }
}

class TelecomStudent extends ITStudent {

    String networkFocus;

    TelecomStudent(String name, int id, String networkFocus) {
        super(name, id);
        this.networkFocus = networkFocus;
    }

    void displayTelecom() {
        showInfo();
        System.out.println("Specialization: Telecom - " + networkFocus);
    }
}

public class ex7_2 {

    public static void main(String[] args) {
        ProgrammingStudent ps = new ProgrammingStudent("Davin", 101, "Java");
        TelecomStudent ts = new TelecomStudent("Rith", 102, "G4 Networks");

        ps.displayProgramming();
        System.out.println();
        ts.displayTelecom();
    }
}
```

Output

```
IT Student Name: Davin, ID: 101
Specialization: Programming in Java

IT Student Name: Rith, ID: 102
Specialization: Telecom - G4 Networks
```

Exercise 8

Lab04.8. Basic Math and Advanced Math (Single Inheritance)

Source Codes

```
class BasicMath {

    double addition(double a, double b) {
        return a + b;
    }

    double subtract(double a, double b) {
        return a - b;
    }
}

class AdvanceMath extends BasicMath {

    double multiply(double a, double b) {
        return a * b;
    }

    double divide(double a, double b) {
        if (b == 0) {
            System.out.println("Error: Infinity!");
            return 0;
        }
        return a / b;
    }
}

public class ex8 {

    public static void main(String[] args) {
        BasicMath bm = new BasicMath();
        System.out.println(bm.addition(6, 9));
        System.out.println(bm.subtract(50000, 36750));

        AdvanceMath am = new AdvanceMath();
        System.out.println(am.addition(6, 9));
        System.out.println(am.subtract(6, 9));
        System.out.println(am.multiply(6, 9));
        System.out.println(am.divide(6, 0));
    }
}
```

Output

```
15.0
13250.0
15.0
-3.0
54.0
```

```
Error: Infinity!
```

```
0.0
```

Link to GitHub Account : [Click Here](#) 

Note: Viewing in VsCode IDE for better formatting!!!