

# Operating Systems Lab 2

Tarek Abdelbari SIBACHIR

January 2025

## 1 Introduction

This lab focuses on learning how to use the Java programming language in a GNU/Linux environment. It involves various coding tasks that test the student's ability to implement and utilize basic Java concepts such as loops, arrays, and user input, along with interacting with the file system.

## 2 Task 1: Introduction to Java Environment

In this task, the instructor provided an overview of the Java software environment, including how to install the Java Development Kit (JDK) and run Java programs on a GNU/Linux system. We learned how to set up the Java environment and tested the installation by running a "Hello, World!" program.

## 3 Task 2: Java Coding Problems

In this section, several coding tasks were provided to be implemented using Java. The tasks involve the use of basic programming concepts like loops, arrays, and string manipulation.

### 3.1 Task 2.1: Guess the Number Game

In this task, we created a Java program where the user is prompted to guess a value between 0 and 100. The user is given 10 attempts, and the program provides feedback on whether the guess is correct or not. If the user guesses correctly, a green "Congrats!" message is displayed. If the user fails after 10 tries, a red "You Lost. Goodbye" message is shown.

Listing 1: Guess the Number Game

```
import java.util.Scanner;

public class GuessTheNumber {
    public static void main(String[] args) {
```

```

Scanner scanner = new Scanner(System.in);
int target = (int) (Math.random() * 101); // Random number between 0 and 100
int attempts = 0;
boolean guessedCorrectly = false;

while (attempts < 10) {
    System.out.print("Guess the number (between 0 and 100): ");
    int guess = scanner.nextInt();
    attempts++;

    if (guess == target) {
        System.out.println("\033[32mCongrats! You guessed the right number.");
        guessedCorrectly = true;
        break;
    } else {
        System.out.println("Wrong guess. Try again.");
    }
}

if (!guessedCorrectly) {
    System.out.println("\033[31mYou Lost. Goodbye.\033[0m"); // Red text
}
}

```

### 3.2 Task 2.2: Student List Using ArrayList

In this task, we created a Java program that maintains a list of students. Each student is represented by their first name, last name, and grade. The user can add students to the list or remove them as needed.

Listing 2: Student List Program

```

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

public class StudentList {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Student> students = new ArrayList<>();

        while (true) {
            System.out.println("Enter 'add' to add a student, 'remove' to remove a student, 'quit' to quit the program.");
            String command = scanner.nextLine();

            if (command.equals("add")) {
                // Add student logic
            }
        }
    }
}

```

```

        System.out.print("Enter first name: ");
        String firstName = scanner.nextLine();
        System.out.print("Enter last name: ");
        String lastName = scanner.nextLine();
        System.out.print("Enter grade: ");
        String grade = scanner.nextLine();
        students.add(new Student(firstName, lastName, grade));
    } else if (command.equals("remove")) {
        System.out.print("Enter first name of the student to remove: ");
        String nameToRemove = scanner.nextLine();
        students.removeIf(student -> student.getFirstName().equals(nameT
    } else if (command.equals("exit")) {
        break;
    }
}

scanner.close();
}
}

class Student {
    private String firstName;
    private String lastName;
    private String grade;

    public Student(String firstName, String lastName, String grade) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.grade = grade;
    }

    public String getFirstName() {
        return firstName;
    }
}

```

### 3.3 Task 2.3: Palindrome Checker

In this task, we created a program that asks the user for a string input and checks whether it is a palindrome (reads the same backward and forward).

Listing 3: Palindrome Checker

```

import java.util.Scanner;

public class PalindromeChecker {

```

```

    public static void main(String [] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string:-");
        String input = scanner.nextLine();
        String reversed = new StringBuilder(input).reverse().toString();

        if (input.equals(reversed)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }
}

```

### 3.4 Task 2.4: Spy Number Checker

The program checks whether a given number is a "spy number." A spy number is a number where the sum of its digits is equal to the product of its digits.

Listing 4: Spy Number Checker

```

import java.util.Scanner;

public class SpyNumberChecker {
    public static void main(String [] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a positive integer:-");
        int number = scanner.nextInt();

        int sum = 0;
        int product = 1;
        int temp = number;

        while (temp > 0) {
            int digit = temp % 10;
            sum += digit;
            product *= digit;
            temp /= 10;
        }

        if (sum == product) {
            System.out.println("The number is a spy number.");
        } else {
            System.out.println("The number is not a spy number.");
        }
    }
}

```

### 3.5 Task 2.5: Create a Directory

In this task, we wrote a program that creates a directory named "Windows" on the user's Desktop directory in GNU/Linux.

Listing 5: Create Directory

```
import java.io.File;

public class CreateDirectory {
    public static void main(String[] args) {
        File dir = new File(System.getProperty("user.home") + "/Desktop/Windows");
        if (!dir.exists()) {
            dir.mkdir();
            System.out.println("Directory 'Windows' created successfully.");
        } else {
            System.out.println("Directory already exists.");
        }
    }
}
```

### 3.6 Task 2.6: Reboot the System

This task involves writing a Java code that reboots the operating system after 5 seconds.

Listing 6: Reboot System

```
public class RebootSystem {
    public static void main(String[] args) throws InterruptedException {
        Thread.sleep(5000); // Wait for 5 seconds
        try {
            String os = System.getProperty("os.name").toLowerCase();
            if (os.contains("win")) {
                Runtime.getRuntime().exec("shutdown -r -t 0");
            } else if (os.contains("nix") || os.contains("nux")) {
                Runtime.getRuntime().exec("reboot");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```