HARAMAYA UNIVERSITY
COLLEGE OF CCI
Department of software engineering
ind. project of object oriented programming (java)

Amanuel Abate----------2122/14

submission date 30/11/2023
submitted to instr..Yohanis Samuel

STUDENTS REGISTRATION SYSTEMS

THIS JAVA PROGRAM IS A SIMPLE STUDENT REGISTRATION SYSTEM FOR HARAMAYA UNIVERSITY.

ArrayList is a dynamic array implementation of the List interface.
Unlike arrays in Java, ArrayList can dynamically resize itself, making it more flexible and convenient to use.

Dynamic Sizing: Automatically adjusts its size as elements are added or removed.
Random Access: Supports fast random access to elements using their index.
Resizable: Unlike arrays, you don't need to define the size in advance.

List is an interface that extends the Collection interface.
It represents an ordered collection of elements and allows duplicate elements.
Being an interface, it provides a common set of methods that any class implementing it should implement, including ArrayList.

It allows users to register new students, display all registered students, and exit the system.

Here's a brief explanation of the code:

Student Registration System Class (StudentRegistrationSystem):

Manages a list of students using **List<Student>** from the **java.util package**.
Provides methods to register a new student and display all registered students.

Main Method:

The main method serves as the entry point of the program.
Uses a Scanner for user input.
It runs in an infinite loop, allowing the user to perform actions until they choose to exit.

Student Class (Student):

Represents a student with various attributes like ID, name, gender, date of birth, etc.
Each student is assigned a unique registration number using a static counter (UID).

Menu:
Displays a menu with three options:

- Register a new student
- Display all registered students
- Exit the system

User Input Handling:
Handles user input for various student details (ID, name, gender, etc.).
Utilizes methods like getValidStringInput to ensure that valid inputs (e.g., names containing only letters) are accepted.
Uses exception handling to handle cases where the user enters incorrect values.

Displaying Students:
Uses a for-each loop to display details of all registered students.
Checks if the list of students is empty before attempting to display them.

Exception Handling:

Catches exceptions (e.g., NumberFormatException) to handle cases where the user enters invalid values, ensuring the program doesn't crash.

Infinite Loop:

The program runs in an infinite loop until the user chooses to exit (case 3).
Invalid choices or exceptions are caught to provide a better user experience.

Overall Flow:

The program guides the user through registering new students or viewing the existing list.
Exception handling is in place to handle unexpected inputs and guide the user to enter correct values.