# ****Student Management System using ArrayList****

**Rujal Baniya**

**Nepal Business College, Affiliated to Lincoln University**

**BIT Second Semester**

**Java Programming**

**Mr. Abhay Das**

# Code Of: ****Student Management System using ArrayList****

*package Projects.Sutdent\_Management\_System;*

*import java.util.ArrayList;*

*import java.util.InputMismatchException;*

*import java.util.Scanner;*

*class Student {*

*private String id;*

*private String name;*

*private int age;*

*private String prg;*

*private String sem;*

*public Student(String id, String name, int age, String prg, String sem) {*

*this.prg = prg;*

*this.sem = sem;*

*this.id = id;*

*this.name = name;*

*this.age = age;*

*}*

*// Getters*

*public String getId() {*

*return id;*

*}*

*public String getName() {*

*return name;*

*}*

*public int getAge() {*

*return age;*

*}*

*public String getPrg() {*

*return prg;*

*}*

*public String getSem() {*

*return sem;*

*}*

*// Setters*

*public void setId(String id) {*

*this.id = id;*

*}*

*public void setName(String name) {*

*this.name = name;*

*}*

*public void setAge(int age) {*

*this.age = age;*

*}*

*public void setPrg(String prg) {*

*this.prg = prg;*

*}*

*public void setSem(String sem) {*

*this.sem = sem;*

*}*

*@Override*

*public String toString() {*

*return "ID=" + id + ",      Name=" + name + ",      Age=" + age + ",        Semester=" + sem + ",       Program=" + prg;*

*}*

*}*

*public class StudentManagementSystem {*

*private ArrayList<Student> students\_list;*

*private Scanner scanner;*

*public StudentManagementSystem() {*

*students\_list = new ArrayList<>();*

*scanner = new Scanner(System.in);*

*}*

*// function to find std in id*

*public Student findStudentById(String id) {*

*for (Student student : students\_list) {*

*if (student.getId().equals(id)) {*

*return student;*

*}*

*}*

*return null;*

*}*

*public void print\_list(ArrayList<Student>  list\_to\_print){*

*for ( Student each\_list: list\_to\_print){*

*System.out.println(each\_list);*

*}*

*}*

*public void search\_std(){*

*System.out.println("\n----------------------- On what Basic You wanna search: -----------------------\n");*

*System.out.println("1. ID");*

*System.out.println("2. Name");*

*System.out.println("3. Age");*

*System.out.println("4. Program");*

*System.out.println("5. Semester");*

*System.out.println("Enter your choice: ");*

*int choice = 0;*

*try{*

*choice = scanner.nextInt();*

*scanner.nextLine(); // Consume newline left-over (input buffer)*

*}*

*catch (InputMismatchException error\_obj){*

*scanner.nextLine();*

*System.out.println("Sorry Only Number are aspectable, "+ error\_obj);*

*search\_std();*

*}*

*switch (choice) {*

*case 1:*

*System.out.println("Enter Student ID to search: ");*

*String id = scanner.nextLine();*

*Student found\_std = findStudentById(id);*

*if (found\_std == null){*

*System.out.println("No Data found...................................................\n");*

*break;*

*}*

*System.out.println(found\_std);*

*break;*

*case 2:*

*System.out.println("Enter Student Name to search: ");*

*String name = scanner.nextLine();*

*ArrayList<Student> found\_std\_list\_name = new ArrayList<>();*

*for (Student each\_std : students\_list)  {*

*if (each\_std.getName().toUpperCase().equals(name.toUpperCase())){*

*found\_std\_list\_name.add(each\_std);*

*}*

*}*

*if (found\_std\_list\_name.isEmpty() ){*

*System.out.println("No Data found...................................................\n");*

*break;*

*}*

*print\_list( found\_std\_list\_name);*

*break;*

*case 3:*

*System.out.println("Enter Student Age to search: ");*

*int age = scanner.nextInt();*

*scanner.nextLine(); // Consume newline left-over*

*ArrayList<Student> found\_std\_list\_age = new ArrayList<>();*

*for (Student each\_std : students\_list)  {*

*if (each\_std.getAge() == age ){*

*found\_std\_list\_age.add(each\_std);*

*}*

*}*

*if (found\_std\_list\_age.isEmpty() ){*

*System.out.println("No Data found...................................................\n");*

*break;*

*}*

*print\_list( found\_std\_list\_age);*

*break;*

*case 4:*

*System.out.println("Enter Student Program to search: ");*

*String prg = scanner.nextLine();*

*ArrayList<Student> found\_std\_list\_prg = new ArrayList<>();*

*for (Student each\_std : students\_list)  {*

*if (each\_std.getPrg().toUpperCase().equals(prg.toUpperCase()) ){*

*found\_std\_list\_prg.add(each\_std);*

*}*

*}*

*if (found\_std\_list\_prg.isEmpty() ){*

*System.out.println("No Data found...................................................\n");*

*break;*

*}*

*print\_list( found\_std\_list\_prg);*

*break;*

*case 5:*

*System.out.println("Enter Student Semester to search: ");*

*String sem = scanner.nextLine();*

*ArrayList<Student> found\_std\_list\_sem = new ArrayList<>();*

*for (Student each\_std : students\_list)  {*

*if (each\_std.getSem().toUpperCase().equals(sem.toUpperCase()) ){*

*found\_std\_list\_sem.add(each\_std);*

*}*

*}*

*print\_list( found\_std\_list\_sem);*

*break;*

*default:*

*System.out.println("Invalid choice. Please try again.");*

*}*

*}*

*// Add a new student*

*public void addStudent() {*

*System.out.println("\n General Information \n");*

*System.out.println("Enter Student ID: ");*

*String id = scanner.nextLine();*

*System.out.println("Enter Student Name: ");*

*String name = scanner.nextLine();*

*System.out.println("Enter Student Age: ");*

*int age = scanner.nextInt();*

*scanner.nextLine(); // Consume newline left-over*

*System.out.println("\n Academic Information \n");*

*System.out.println("Enter Student Program: ");*

*String prg = scanner.nextLine();*

*System.out.println("Enter Student Semester: ");*

*String sem = scanner.nextLine();*

*Student student = new Student(id, name, age,prg, sem);*

*students\_list.add(student);*

*System.out.println("Student added successfully!");*

*}*

*// Remove a student by ID*

*public void removeStudent() {*

*System.out.println("Enter Student ID to remove: ");*

*String id = scanner.nextLine();*

*Student removed\_student = findStudentById(id);*

*boolean removed = students\_list.removeIf(student -> student.getId().equals(id));*

*if (removed) {*

*System.out.println("Student :" + removed\_student);*

*System.out.println("Removed successfully!");*

*} else {*

*System.out.println("Student with ID " + id + " not found.");*

*}*

*}*

*// Display all students*

*public void displayStudents() {*

*if (students\_list.isEmpty()) {*

*System.out.println("No students to display.");*

*} else {*

*for (Student student : students\_list) {*

*System.out.println(student);*

*}*

*}*

*}*

*// Main menu*

*public void menu() {*

*while (true) {*

*System.out.println("\n--- Student Management System ---");*

*System.out.println("1. Add Student");*

*System.out.println("2. Remove Student");*

*System.out.println("3. Display All Students");*

*System.out.println("4. Search Students, with Filter");*

*System.out.println("5. Exit");*

*System.out.print("Choose an option: ");*

*int choice = 0;*

*try{*

*choice = scanner.nextInt();*

*scanner.nextLine(); // Consume newline left-over (input buffer)*

*}*

*catch (InputMismatchException error\_obj){*

*scanner.nextLine();*

*System.out.println("Sorry Only Number are aspectable, "+ error\_obj);*

*continue;*

*}*

*switch (choice) {*

*case 1:*

*addStudent();*

*break;*

*case 2:*

*removeStudent();*

*break;*

*case 3:*

*displayStudents();*

*break;*

*case 4:*

*search\_std();*

*break;*

*case 5:*

*System.out.println("Exiting...");*

*return;*

*default:*

*System.out.println("Invalid choice. Please try again.");*

*}*

*}*

*}*

*public static void main(String[] args) {*

*StudentManagementSystem sms = new StudentManagementSystem();*

*sms.menu();*

*}*

*}*

# ****What I learned: Student Management System using ArrayList****

#### Reflections on Building This Project:

1. **Hands-On Java Practice:** This project marked my first time practicing Java programming on a computer, having previously only worked on it in theory.
2. **Understanding the ArrayList Concept:** Gained a solid understanding of ArrayList and its practical applications, with the help of tutorials from W3Schools.
3. **Clearer Grasp of Java Syntax:** Improved my understanding of Java syntax, making my coding process smoother and more efficient.
4. **Exploring Object-Oriented Programming (OOP):** Experimented with classes, inheritance, and other OOP principles, enhancing my grasp of object-oriented design.
5. **Mastering Exception Handling:** Learned how to use try and catch blocks effectively to handle exceptions and maintain robust code.
6. **Deepening Knowledge of Access Modifiers:** Gained a clearer understanding of Java access modifiers like public and private, and how they control access to different parts of a program.
7. **Overriding the** toString() **Method:** Discovered how to override the toString() method to provide a custom string representation for objects. For example:

*@Override*

*public String toString() {*

*return "Custom description here";*

*}*

1. **Using Functions Professionally:** Developed skills in designing and implementing functions effectively to promote clean, modular code.
2. **Passing Parameters and Using ArrayLists:** Learned how to pass parameters, including ArrayList objects, to methods and utilize them effectively in various scenarios.