import java.util.\*;

public class NewClass7 {

private static ArrayList<Student> students = new ArrayList<>();

public static void main(String[] args) {

while (true) {

try {

System.out.println();

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

System.out.println(" STUDENT MANAGEMENT SYSTEM ");

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

System.out.println();

System.out.println("1.Add a new student ");

System.out.println("2.Search for a student's address");

System.out.println("3.Show all students");

System.out.println("4.Add course for a student");

System.out.println("5.Search for a students courses");

System.out.println("6.Delete a student");

System.out.println("7.Add a mentor");

System.out.println("8.Search a mentor");

System.out.println("9.All students under (faculty information)");

System.out.println("10.Drop course");

System.out.println("11.Exit");

Scanner input = new Scanner(System.in);

System.out.println("Choose an option: ");

int n = input.nextInt();

input.nextLine();

switch (n) {

case 1: //Adding new student

System.out.print("Enter student name: ");

String sname = input.nextLine();

System.out.print("Enter student id: ");

String sId = input.nextLine();

System.out.print("Enter area name: ");

String area = input.nextLine();

System.out.print("Enter Area code: ");

int code = input.nextInt();

Address adrs = new Address(area, code);

addstudent(sname, sId, adrs);

input.nextLine();

break;

case 2: // Searching for a student address

System.out.print("Enter student id: ");

String id1 = input.nextLine();

searchstudent(id1);

break;

case 3: // Basic information of student

showallBasicInformation();

break;

case 4: //Adding course

System.out.print("Enter student id: ");

String stuId2 = input.nextLine();

addcourse(stuId2);

break;

case 5: //Searching for added coures

System.out.print("Enter student id: ");

String stuId = input.nextLine();

searchStudentCourse(stuId);

break;

case 6: //Deleting whole information of a student

System.out.println("Enter student id: ");

String stuId1 = input.nextLine();

delete(stuId1);

System.out.println("Successfully deleted. ");

break;

case 7: // Adding mentor

System.out.println("Enter student id : ");

String stuID = input.nextLine();

System.out.println("Enter teacher name : ");

String tcherName = input.nextLine();

System.out.println("Enter teacher id: ");

String tcherId = input.nextLine();

Teacher t = new Teacher(tcherName, tcherId);

addt(stuID, t);

break;

case 8: //Searching for a mentor

System.out.println("Enter student id: ");

String studntId = input.nextLine();

srch(studntId);

break;

case 9:// mentor information

System.out.println("Enter Teacher id: ");

String techrID = input.nextLine();

teachersstudents(techrID);

break;

case 10: // withdrawing course

System.out.println("Enter Student id");

String studId = input.nextLine();

System.out.println("Enter course name: ");

String crsename = input.nextLine();

System.out.println("Enter course code: ");

int courseID = input.nextInt();

drop(courseID, crsename, studId);

break;

case 11:

System.out.println();

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

System.out.println("Exiting the program.Goodbye!!");

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

System.out.println();

System.exit(0);

default:

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

}

} catch (Exception e) {

System.out.println("Wrong input. Please try again!!");

}

}

}

private static void addstudent(String name, String id, Address a) {

Student newStudent = new Student(name, id, a);

students.add(newStudent);

System.out.println("New student added successfully:");

}

private static void searchstudent(String studentId) {

boolean found = false;

for (Student student : students) {

if (student.getId().equals(studentId)) {

System.out.println("Student found:");

System.out.println(student.toString());

System.out.println(student.getAddress().toString());

found = true;

break;

}

}

if (!found) {

System.out.println("not found");

}

}

public static void showallBasicInformation() {

for (Student y : students) {

System.out.println("Show all student details: ");

System.out.println(y.toString());

System.out.println(y.getAddress().toString());

}

}

public static void addcourse(String id) {

Scanner input = new Scanner(System.in);

for (Student g : students) {

if (g.getId().equals(id)) {

while (true) {

System.out.print("No. of courses (3 - 6): ");

int n = input.nextInt();

if (n >= 3 && n <= 6) {

for (int i = 1; i <= n; i++) {

System.out.println("Enter course name");

String coursename = input.nextLine();

System.out.println("Enter coursecode");

int coursecode = input.nextInt();

Course c = new Course(coursename, coursecode);

g.setList(c);

}

break;

} else {

System.out.println("Try again.");

}

}

}

}

}

public static void searchStudentCourse(String id) {

for (Student t : students) {

if (t.getId().equals(id)) {

for (Course i : t.getList()) {

System.out.println(i.getCoursename()+ "\t");

System.out.println(i.getCoursecode());

}

}

}

}

public static void delete(String n) {

int i = -1;

for (Student s : students) {

i++;

if (s.getId().equals(n)) {

break;

}

}

students.remove(i);

}

public static void addt(String id, Teacher teacher) {

boolean t=false;

for (Student s : students) {

if (s.getId().equals(id)) {

s.setTeacher(teacher);

t=true;

break;

}

}

if(t==true)

{

System.out.println("mentor added succesfully");

}

if(t==false)

{

System.out.println("student not found");

}

}

public static void srch(String id) {

for (Student j : students) {

if (j.getId().equals(id)) {

System.out.println(j.getTeacher().toString());

}

}

}

public static void teachersstudents(String teachersId) {

for (Student s : students) {

if (s.getTeacher().getId().equals(teachersId)) {

System.out.println(s.getTeacher().getName());

break;

}

}

for (Student s : students) {

if (s.getTeacher().getId().equals(teachersId)) {

System.out.println(s.toString());

}

}

}

public static void drop(int courseId, String name, String studid) {

boolean t=false;

int i = -1;

for (Student s : students) {

if (s.getId().equals(studid)) {

for (Course c : s.getList()) {

i++;

if (c.getCoursecode() == (courseId) && c.getCoursename().equals(name)) {

s.dropcourse(i);

t=true;

break;

}

}

}

}

if(t==true)

{

System.out.println("course dropped succesfully");

}

if(t==false)

{

System.out.println("student not found");

}

}

}

class Student {

private String name;

private Teacher teacher;

private String id;

private Address address;

private ArrayList<Course> list = new ArrayList<>();

public Student(String name, String id, Address address) {

this.name = name;

this.id = id;

this.address = address;

}

@Override

public String toString() {

return "Student Name:" + name + " Student ID:" + id;

}

public Address getAddress() {

return address;

}

public void setAddress(Address address) {

this.address = address;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public ArrayList<Course> getList() {

return list;

}

public void setList(Course c) {

list.add(c);

}

public Teacher getTeacher() {

return teacher;

}

public void setTeacher(Teacher teacher) {

this.teacher = teacher;

}

public void dropcourse(int s) {

list.remove(s);

}

}

class Address {

private String area;

private int code;

public Address(String area, int code) {

this.area = area;

this.code = code;

}

@Override

public String toString() {

return "Area Name: " + area + " Area Code:" + code;

}

}

class Course {

private String coursename;

private int coursecode;

public Course(String coursename, int coursecode) {

this.coursename = coursename;

this.coursecode = coursecode;

}

public String getCoursename() {

return coursename;

}

public void setCoursename(String coursename) {

this.coursename = coursename;

}

public int getCoursecode() {

return coursecode;

}

public void setCoursecode(int coursecode) {

this.coursecode = coursecode;

}

}

class Teacher {

private String name;

private String id;

public Teacher() {

}

public Teacher(String name, String id) {

this.name = name;

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

@Override

public String toString() {

return "Teacher Name:" + name + " Teacher Id no: " + id;

}

}