```
::::::::::::::
Lab0Status.txt
::::::::::::::
Problem 1: compiles, runs correctly on all provided input

Update 1: Updated header of driver to align with requirements. Also fixed formatti
ng in Driver for the same purpose::::::::::::::
Lab0Conclusions.txt
::::::::::::::
Question 1: This problem was a lab from OOPDA that involved utilizing polymorphism
 to create a Person object that will be the 'skeleton' for Instructor and Student
objects. My solution is efficient because not only does it reflect polymorphism, b
ut lambda expressions and regular expressions are used for the best possible effic
iency and memory optimization.

Question 2: From a thorough look of the course site and materials, it seems that s
trong foundational knowledge from the prior two courses (IOOP and OOPDA) are key i
n being successful in DSA. Furthermore, applying one self to study new material li
ke reading the textbook and practicing outside of class will only strengthen the l
ikelihood of being successful. It all boils down to hard work and dedication. ::::
::::::::::
Lab0Driver.java
::::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab X Problem Y [instantiate X and Y]
 * Status: Incomplete
 * Last update: 01/17/23
 * Submitted:  01/19/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.01.18
 */

package Person;

import java.util.HashMap;
import java.util.Scanner;

public class Driver {
    public String department;
    public String major;
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        HashMap<Integer, Person> map = new HashMap<Integer, Person>();

        System.out.println("How should we evaluate? [Enter 'oldest' or 'youngest']
");
        String ageLabel = input.nextLine();

        System.out.println("Enter person's first name: " + "\n");
        String firstName = input.nextLine();

        System.out.println("Enter person's middle name: " + "\n");
        String middleName = input.nextLine();

        System.out.println("Enter person's last name: " + "\n");
        String lastName = input.nextLine();

        System.out.println("Enter person's email address" + "\n");
        String emailTest = input.nextLine();
        System.out.println("Enter person's SSN (in ###-##-#### format): " + "\n");
        String ssnTest = input.nextLine();

        System.out.println("Enter person's age: " + "\n");
        String ageTest = input.nextLine();
        input.close();
        map.put(1, new Person(916421999, "Rip", "Van", "Winkle", "rip@rowan.edu",
"111-22-3333", 99));
        /*
         * 916421999 Rip Van Winkle (Person) rowan.edu 3333 oldest: true
         */
        Person student = new Student(916422000, "Jane", "Marie", "Doe", "jmd@stude
nts.rowan.edu", "222-33-4444", 17, "Computer Science");
        /*
         * 916422000 Jane Marie Doe (Student) students.rowan.edu 4444 oldest: fals
e
         */
        Person instructor = new Instructor(916422001, "Chia", "C.", "Chien", "chie
n@rowan.edu", "333-44-5555", 40, "Math/Science");
        /*
         * 916422001 Chia C. Chien (Instructor) rowan.edu 5555 oldest: false
         */
        map.put(2, student);
        map.put(3, instructor);

        Integer age = 0;
        if (Person.isAgeValid(ageTest)) {
            age = Integer.parseInt(ageTest);
        }

        String email = null;
        if (Person.isEmailValid(emailTest)) {
            email = emailTest;
        }
        email = emailTest;

        String ssn = null;
        if (Person.isSSNValid(ssnTest)) {
            ssn = ssnTest;
        }
        ssn = ssnTest;
        input.close();

        map.put(4, new Person(916422002, firstName, middleName, lastName, email, s
sn, age));
        if (ageLabel.matches("Youngest") || ageLabel.matches("youngest")) {
            for (int key : map.keySet()) // iterate thru HashMap to label correct
youngest
            {
                String label = ""; // empty because there's no "youngest" yet
                map.get(key);
                if (Person.youngestP.test(map.get(key))) {
                    label = "Youngest: true";
                } else {
                    label = "Youngest: false";
                }

                System.out.println("\n" + map.get(key).getId()
                                   + "\n" + map.get(key).toString()
                                   + "\n" + map.get(key).getEmailDomain()
                                   + "\n" + map.get(key).getLast4SSN()
                                   + "\n" + label
                                   + "\n" + map.get(key).getMajor()
                                   + "\n" + map.get(key).getDepartment());
            }
        } else if ((ageLabel.matches("Oldest") || ageLabel.matches("oldest"))) {
            for (int key : map.keySet()) // iterate thru HashMap to label correct
youngest
            {
                String label = ""; // empty because there's no "youngest" yet
                map.get(key);
                if (Person.oldestP.test(map.get(key))) {
                    label = "Oldest: true";
                } else {
                    label = "Oldest: false";

                }

                System.out.println("\n" + map.get(key).getId()
                                   + "\n" + map.get(key).toString()
                                   + "\n" + map.get(key).getEmailDomain()
                                   + "\n" + map.get(key).getLast4SSN()
                                   + "\n" + label
                                   + "\n" + map.get(key).getMajor()
                                   + "\n" + map.get(key).getDepartment());
            }

        } else {
            System.out.println("Error: Invalid input. Type 'youngest' or 'oldest'"
);
        }
    }
}
::::::::::::::
Lab0SampleRuns.txt
::::::::::::::
How should we evaluate? [Enter 'oldest' or 'youngest']
youngest
Enter person's first name:

Antonio
Enter person's middle name:

N/A
Enter person's last name:

Rosado
Enter person's email address

rosado44@students.rowan.edu
Enter person's SSN (in ###-##-#### format):

555-55-6575
Enter person's age:

20

916421999
Rip Van Winkle
rowan.edu
3333
Youngest: false

916422000
Jane Marie Doe
students.rowan.edu
4444
Youngest: true
Computer Science


916422001
Chia C. Chien
rowan.edu
5555
Youngest: false

Math/Science

916422002
Antonio N/A Rosado
students.rowan.edu
6575
Youngest: false
```