Manager

// addStudent(), addInstructor(), addCourse(), addAssignment(), enrollStudent() // getStudent(), getInstructor, getCourse, getStudentGrade() // printStudents(), printStudent(), exportStudents(), printInstructors(), printCourses()

UserInterface

private Manager manager private Scanner scanner

public UserInterface(Manager manager)
 public void start()
 private void welcomeMessage()
 private void mainMenu()
 private void manageStudents()
 private void manageAssignments()
 private void manageInstructors()
 private void manageCourses()
 private void exportData()

Course

private String courseName private String instructor private String roomNumber private String schedule private ArrayList<Student> roster

Grade

private int grade;

public void setGrade(int grade)
public int getGrade()
public String
getLetterGrade(ArrayList<Assignment>
assignments)
public void getClassGrades

Assignment

private String assignmentName private int score private static final int MAX_SCORE = 100

public Assignment(String assignmentName, int score)

public void setAssignmentName(string assignmentName)

public void setScore(int score)

public String getAssignmentName()

public int getMaxScore()

public int qetScore()

public void printDetails()

Abstract Person

private String name private String email

public Person(String name, String email)
 public void printDetails()
 public void setName(String name)
 public String getName()
 public void setEmail(String email)
 public String getEmail()

Student (extends Person)

public int studentID
private ArrayList<Assignment> assignments;

public Student(String name, String email, int studentID)
public void setID(int ID)
public int getID()
public void addAssignment(String assignmentName, int score)
public ArrayList<Assignment> getAssignments()
public void exportStudent()

Instructor (extends Person)

private String courseName;

public Instructor(String name, String email, String courseName) public String getCourseName() public void setCourseName() public void printDetails()

Polymorphism

Student and Instructor have both inherited methods (IE printDetails()) but will be overridden to behave differently for each class

Encapsulation

Frequent use of setters and getters with private variables and public methods

Inheritance

Student and Instructor both inherit Person

Abstraction

Explanation

There is an abstract class Person and a console interface to hide complexity

Description & Justification

Using a manager class to handle all objects, as well as dynamic ArrayLists to contain them. An interface is used by the user, which itself makes use of the manager. This makes it simple to interact with all of the classes.