Manager

public void addStudent(String name, String email, int ID)
public int getStudent(String name);
public void addInstructor(String name, String email, String
course)
public int getInstructor(String name);
public void addCourse(String name, String instructor, String

roomNumber)

public int getCourse(String name);

Course

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

Grade (Static?)

public String getLetterGrade()
public void getClassGrades()
public void getStudentGrades()

Assignment

private String name private String course private int maxGrade private ArrayList<Integer> studentIDs private ArrayList<Integer> studentGrades

public Assignment(String name, String course)
 public void printDetails()
 public void setName(String name)
 public String getName()
 public void setMaxGrade(int max)
 public int getMaxGrade()
 public void setStudent(int ID, int grade)
 public int getStudentGrade()
 public void setCourse(String course)
 public String getCourse()

Person

private String name private String email

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

-Student & Instructor Inherit Person-

Student

public int studentID

Instructor

private String courseName

public Instructor(String name, String, email, String course) public void printDetails()

Polymorphism

Student and Instructor both have methods with the same names (EX setName() or printDetails()) but they will be executed differently

Encapsulation

Frequent use of setters and getters rather than directly manipulating variables

Inheritance

Student and Instructor both inherit Person

Abstraction

Explanation

Person has an abstract method printDetails() that will be implemented in Student and Instructor

Description & Justification

Using a manager class with to handle all objects, as well as dynamic ArrayLists to contain them. This makes it simple to interact with all of the classes.