Manager

private ArrayList<Student> students = new ArrayList<Student>()
private ArrayList<Instructor> instructors = new ArrayList<Instructor>()
private ArrayList<Course> courses = new ArrayList<Course>()

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

public void printDetails()
public void setCourseName(String course)
public String getCourseName()
public void setInstructor(String instructor)
public String getInstructor()
public void setRoomNumber(String num)
public String getRoomNumber()
public void setSchedule(String schedule)
public String getSchedule()

Assignment

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

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()

Person

private String name
private 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—

Grade (Static?)

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

Student

public int studentID

public void printDetails()
public void exportDetails();
public void setID(int ID)
public int getID()
public void exportStudent()

Instructor

private String courseName

public void printDetails()

Explanation

Polymorphism	Encapsulation	Inheritance	Abstraction	Description & Justification
Student and Instructor both have methods with the same names (EX setName() or printDetails()) but they will be executed differently	Frequent use of setters and getters rather than directly manipulating variables	Student and Instructor both inherit Person	Person has an abstract class getDetails() that will be implemented in Student and Instructor	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.