## Manager

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

## Interface

// TDB

## Course

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

public Course(//all parameters but roster)
// setters for courseName, roomNumber,
instructor, schedule
// getters for courseName, roomNumber,
instructor, schedule, roster
public void setRoster(ArrayList<Student> roster)
public void addStudent(Student student)
public void printStudentRoster()
public void printDetails()

## 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 maxScore = 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 getScore() 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 printDetails()

# Instructor (extends Person)

private String courseName;

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

#### Explanation **Polymorphism Encapsulation** Inheritance **Abstraction Description & Justification** Abstract class Person inherited by Student and Instructor both have Frequent use of setters and getters Student and Instructor both inherit Using a manager class with to handle all objects, as well as dynamic ArrayLists to contain them. methods with the same names (EX rather than directly manipulating Person Student and Instructor This makes it simple to interact with all of the classes. setName() or printDetails()) but they variables will be executed differently