**Homework 7: People at College**

1. Objective

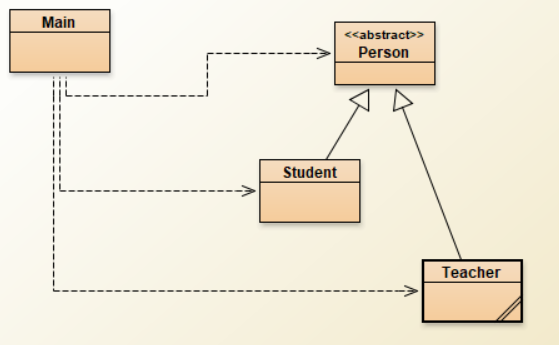
In this project, you’ll create several classes using inheritance and polymorphism. This will require you to create abstract method and class. There won’t be any exciting UI here, just some output of array data.

1. Structure
2. There are two types of College people: **Students** and **Teachers**.
3. All people have an **ID**, **Name**, and **Email**.
4. All IDs have validation rules, but rules differ:
   * Students – 9 digits;
   * Teachers – 6 alphanumeric.
5. For Students, we want to maintain **a list of courses taken** (up to 30 courses) and **associated grades**; we want to obtain overall **grade averages** as well.
6. For Teachers, we **record courses** they have taught (up to 50); we want a way to add courses to that list.

|  |
| --- |
| **Person** |
| -id : String |
| -name : String |
| -email : String |
| **Constructors** |
| +Person(id : String, name: String)  +Person(id : String, name: String, email : String ) |
| **Accessors** |
| +getId() : String |
| +getName() : String |
| +getEmail() : String |
| **Mutators**  +setId(id : String)  +setName(name: String)  +setEmail(email : String) |
| **Other Methods** |
| +isValidId(id : String) : Boolean |
| +toString() : String |
| +test() |

\*\* isValidId() method should be an abstract.

**An UML class diagram**



|  |
| --- |
| **Students** |
| -courseTaken : String[30] |
| -courseGrades : Double[30] |
| -nextCourseIndex : Integer |
| **Constructor** |
| + Students (id : String, name: String, email : String ) |
| **Methods** |
| +isValidIdId(id : String) : Boolean |
| +courseCompletion(courseName : String, courseGrade : Double) |
| +getAverageGrade() : Double |
| +toString() : String |
| +test() |

|  |
| --- |
| **Teachers** |
| -coursesTaught: String[50] |
| -nextCourseIndex : Integer |
| **Constructor** |
| + Students (id : String, name: String, email : String ) |
| **Methods** |
| +isValidIdId(id : String) : Boolean |
| +addCourseTaught (courseName : String) |
| +toString() : String |
| +test() |

1. Code Implementation

Create a **main** class which creates a new instance of the **Students** and **Teachers** classes and displays all information. See examples of output.

1. Testing

* Write test methods for each of your methods.
* Test with different courses and grades to make sure your algorithms are correct.
* Manual inspection will be necessary; the debugger is your friend and will be very useful here.

1. Submitting Your Work

* Submit your .jar or .zip file
* If your test method relies on test files, be sure and include them in your submission.

Examples of Output

