

---

**Computer Science 2212b**  
**Team 7**  
**Assignment #2**

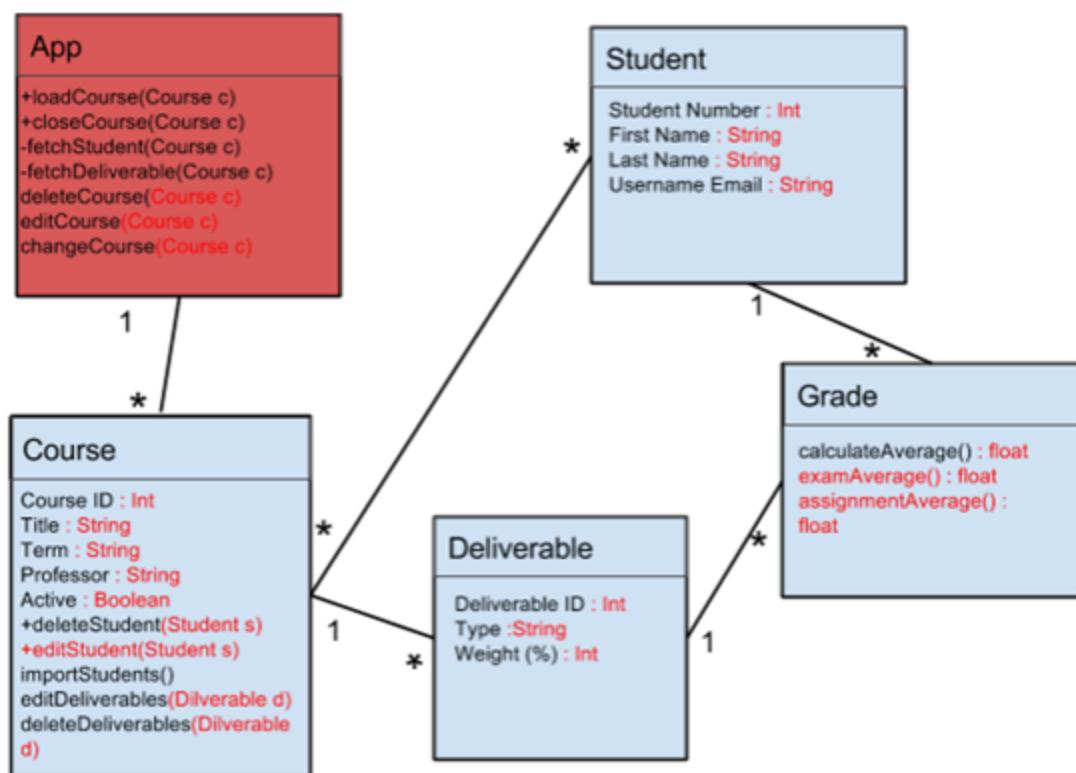
February 28th, 2014

**Nick Delben**  
**Ben Dumais**  
**Alex Gyori**  
**Albert Van Santvoort**  
**Holly Voysey**

---

### Revised UML

This is our previous UML diagram with changes, any new features or corrections made are in red.



### Revised User Stories

<https://www.pivotaltracker.com/s/projects/1004174>



Since report one, 9 user stories have been implemented:

- Create Course
- Edit Course
- Delete Course
- Create Student
- Edit Student
- Delete Student
- Create Deliverable
- Edit Deliverable
- Delete Deliverable

These stories can be tested and used by building and running the package: The gui allows users to switch between tabs for each class where existing objects are displayed in a table that also allows editing of data, and also allows for creation and deletion of objects through the buttons and fields on the side.

cs2212.team7

## Class Course

java.lang.Object  
cs2212.team7.Course

```
public class Course
extends java.lang.Object
```

The class represents a single course in the gradebook. Each course has a number of deliverables and students. Each course can be identified by a name, a course code, and a term.

**Author:**

Alex Gyori, #250643023

### Constructor Summary

**Constructors****Constructor and Description**

|  |
|--|
| <code>Course(java.lang.String name, java.lang.String term, java.lang.String code)</code> |
| Constructor for class Course   |

### Method Summary

**Methods**

| Modifier and Type                 | Method and Description   |
|-----------------------------------|--|
| void                              | <code>addDeliverable(Deliverable newDeliverable)</code><br>Adds a deliverable to the course. |
| void                              | <code>addStudent(long newStudentId)</code><br>Adds a student to the course                   |
| java.lang.String                  | <code>getCode()</code><br>Returns the course code  |
| float                             | <code>getCourseAverage()</code><br>Calculates and returns the average grade of the course    |
| java.util.LinkedList<Deliverable> | <code>getDeliverables()</code><br>Returns the list of deliverables the course contains       |
| java.lang.String                  | <code>getName()</code><br>Returns the name of the course                                     |

|                                      |  |
|--------------------------------------|--|
| float                                | <b>getStudentAverage</b> (int targetId)<br>Calculates and returns the average grade of a single student for this course. |
| java.util.LinkedList<java.lang.Long> | <b>getStudentIds</b> ()<br>Returns the list of IDs for the students enrolled in the course                               |
| java.lang.String                     | <b>getTerm</b> ()<br>Returns the term the course will be taught in   |
| void                                 | <b>removeDeliverable</b> (java.lang.String targetDelName)<br>Removes a deliverable from the course                       |
| void                                 | <b>removeStudent</b> (int targetId)<br>Removes a student from the course   |
| void                                 | <b>setCode</b> (java.lang.String code)<br>Sets the course code   |
| void                                 | <b>setName</b> (java.lang.String name)<br>Sets the name of the course  |
| void                                 | <b>setTerm</b> (java.lang.String term)<br>Sets the term of the course  |

## Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### Course

```
public Course(java.lang.String name,
             java.lang.String term,
             java.lang.String code)
```

Constructor for class Course

#### Parameters:

- name - Name of the course
- term - Term the course will be being taught in
- code - Course code

## Method Detail

### setName

```
public void setName(java.lang.String name)
```

Sets the name of the course

**Parameters:**

name - The new name of the course

### **setTerm**

```
public void setTerm(java.lang.String term)
```

Sets the term of the course

**Parameters:**

term - The new term of the course

### **setCode**

```
public void setCode(java.lang.String code)
```

Sets the course code

**Parameters:**

code - The new course code

### **addDeliverable**

```
public void addDeliverable(Deliverable newDeliverable)
```

Adds a deliverable to the course.

**Parameters:**

newDeliverable - The new deliverable to be added

### **removeDeliverable**

```
public void removeDeliverable(java.lang.String targetDelName)
```

Removes a deliverable from the course

**Parameters:**

targetDelName - The name of the deliverable to be removed

### **addStudent**

```
public void addStudent(long newStudentId)
```

Adds a student to the course

**Parameters:**

newStudentId - The ID of the student to be added

**removeStudent**

```
public void removeStudent (int targetId)
```

Removes a student from the course

**Parameters:**

targetId - The ID of the student to be removed

**getCourseAverage**

```
public float getCourseAverage ()
```

Calculates and returns the average grade of the course

**Returns:**

The average grade of the course

**getStudentAverage**

```
public float getStudentAverage (int targetId)
```

Calculates and returns the average grade of a single student for this course. If the student does not have a mark in a deliverable it is considered a 0.

**Parameters:**

targetId - The ID of the target student

**Returns:**

The average grade for the target student

**getName**

```
public java.lang.String getName()
```

Returns the name of the course

**Returns:**

The name of the course

**getTerm**

```
public java.lang.String getTerm()
```

Returns the term the course will be taught in

**Returns:**

The term of the course

### **getCode**

```
public java.lang.String getCode()
```

Returns the course code

**Returns:**

The course code

### **getDeliverables**

```
public java.util.LinkedList<Deliverable> getDeliverables()
```

Returns the list of deliverables the course contains

**Returns:**

### **getStudentIds**

```
public java.util.LinkedList<java.lang.Long> getStudentIds()
```

Returns the list of IDs for the students enrolled in the course

**Returns:**

list of IDs for the students enrolled in the course

cs2212.team7

## Class Deliverable

java.lang.Object  
cs2212.team7.Deliverable

```
public class Deliverable
extends java.lang.Object
```

### Constructor Summary

#### Constructors

##### Constructor and Description

|   |
|---|
| <code>Deliverable(java.lang.String dName, int dWeight, boolean dB)</code> |
|---|

### Method Summary

#### Methods

| Modifier and Type | Method and Description   |
|-------------------|--|
| void              | <code>addGrade(long targetId, float newGrade)</code><br>Adds a grade to the deliverable                        |
| boolean           | <code>containsStudent(long targetId)</code><br>Checks to see if the student has a mark in this deliverable.    |
| float             | <code>getAverage()</code><br>Calculates and returns the average of the deliverable across all students         |
| java.lang.String  | <code>getName()</code>   |
| float             | <code>getStudentGrade(long targetId)</code><br>Returns the grade of a particular student for this deliverable. |
| int               | <code>getWeight()</code>   |
| boolean           | <code>isEmpty()</code><br>Checks to see whether there are grades in this deliverable                           |
| boolean           | <code>isExam()</code>  |
| void              | <code>removeGrade(long targetId)</code><br>Removes a grade from the deliverable.                               |
| void              | <code>setExam(boolean b)</code>  |
| void              | <code>setGrade(long targetId, float newMark)</code>  |

Changes the mark of an existing grade in the deliverable.

```
void          setName (java.lang.String dName)
void          setWeight(int dWeight)
```

#### Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

### Constructor Detail

#### Deliverable

```
public Deliverable(java.lang.String dName,
                  int dWeight,
                  boolean dB)
```

### Method Detail

#### getName

```
public java.lang.String getName()
```

#### getWeight

```
public int getWeight()
```

#### isExam

```
public boolean isExam()
```

#### setName

```
public void setName(java.lang.String dName)
```

#### setWeight

```
public void setWeight(int dWeight)
                      throws DeliverableException
```

**Throws:**

```
DeliverableException
```

**setExam**

```
public void setExam(boolean b)
```

**addGrade**

```
public void addGrade(long targetId,  
                     float newGrade)  
    throws DeliverableException
```

Adds a grade to the deliverable

**Parameters:**

targetId - The ID number of the student whose grade is being added

newGrade - The mark of the student to be added

**Throws:**

DeliverableException - The student already has a grade for this deliverable

**removeGrade**

```
public void removeGrade(long targetId)  
    throws DeliverableException
```

Removes a grade from the deliverable.

**Parameters:**

targetId - The ID number of the student whose grade is to be removed

**Throws:**

DeliverableException - The student could not be found in this deliverable

**setGrade**

```
public void setGrade(long targetId,  
                     float newMark)
```

Changes the mark of an existing grade in the deliverable.

**Parameters:**

targetId - The ID of student whose mark is to be changed

newMark - The new mark of the student

## **getAverage**

```
public float getAverage()
```

Calculates and returns the average of the deliverable across all students

**Returns:**

The average mark scored on the deliverable

## **getStudentGrade**

```
public float getStudentGrade(long targetId)
    throws DeliverableException
```

Returns the grade of a particular student for this deliverable.

**Parameters:**

targetId - The ID of the student whose mark we seek

**Returns:**

The mark of the student for this deliverable

**Throws:**

DeliverableException - The student could not be found

## **isEmpty**

```
public boolean isEmpty()
```

Checks to see whether there are grades in this deliverable

**Returns:**

true if there are no grades in this deliverable

## **containsStudent**

```
public boolean containsStudent(long targetId)
```

Checks to see if the student has a mark in this deliverable.

**Parameters:**

targetId - The ID of the student we are seeking

**Returns:**

true if the student exists in the deliverable

## Class Grade

java.lang.Object  
Grade

```
public class Grade  
extends java.lang.Object
```

This class represents a single grade a student has for a particular deliverable. The Grade object stores the ID of the student and the actual mark.

### Author:

Alex Gyori, #250643023

## Constructor Summary

### Constructors

#### Constructor and Description

**Grade**(long studentId, float mark)

Constructor for class Grade

## Method Summary

### Methods

| Modifier and Type | Method and Description   |
|-------------------|--|
| float             | <b>getMark</b> ()<br>Gets the mark of the student                                      |
| long              | <b>getStudentId</b> ()<br>Gets the student ID  |
| void              | <b>setGrade</b> (float newMark)<br>Changes the mark of the student for this grade only |
| void              | <b>setStudentId</b> (long newId)<br>Changes the ID pertaining to this grade.           |

#### Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### Grade

```
public Grade(long studentId,  
            float mark)
```

Constructor for class Grade

#### Parameters:

studentId - The ID of the student  
mark - The mark the student receives on the deliverable

## Method Detail

### getStudentId

```
public long getStudentId()
```

Gets the student ID

#### Returns:

The student ID

### getMark

```
public float getMark()
```

Gets the mark of the student

#### Returns:

The mark of the student

### setGrade

```
public void setGrade(float newMark)
```

Changes the mark of the student for this grade only

#### Parameters:

newMark - The new mark of the student

### setStudentId

```
public void setStudentId(long newId)
```

Changes the ID pertaining to this grade.

**Parameters:**

newId -

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

[Summary](#): [Nested](#) | [Field](#) | [Constr](#) | [Method](#)      [Detail](#): [Field](#) | [Constr](#) | [Method](#)

---

cs2212.team7

## Class Student

java.lang.Object  
cs2212.team7.Student

```
public class Student
extends java.lang.Object
```

### Constructor Summary

#### Constructors

##### Constructor and Description

|   |
|---|
| <b>Student</b> (java.lang.String first, java.lang.String last, java.lang.String mail, long num)<br>Constructor for student object |
|---|

### Method Summary

#### Methods

| Modifier and Type | Method and Description  |
|-------------------|---|
| java.lang.String  | <a href="#"><b>getEmail</b></a> ()<br>Get for student email                   |
| java.lang.String  | <a href="#"><b>getFName</b></a> ()<br>Get for first name                      |
| java.lang.String  | <a href="#"><b>getLName</b></a> ()<br>Get for last name                       |
| long              | <a href="#"><b>getStudentNum</b></a> ()<br>Get for student number             |
| void              | <a href="#"><b>setEmail</b></a> (java.lang.String email)<br>Set for email     |
| void              | <a href="#"><b>setFName</b></a> (java.lang.String name)<br>Set for first name |
| void              | <a href="#"><b>setLName</b></a> (java.lang.String name)<br>Set for last name  |
| void              | <a href="#"><b>setStudentNum</b></a> (long num)<br>Set for student number     |

## Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Constructor Detail

### Student

```
public Student(java.lang.String first,  
              java.lang.String last,  
              java.lang.String mail,  
              long num)
```

Constructor for student object

**Parameters:**

first - String for first name  
last - String for last name  
mail - String for email address  
num - Long for student number

## Method Detail

### getFName

```
public java.lang.String getFName()
```

Get for first name

**Returns:**

String of student first name

### getLName

```
public java.lang.String getLName()
```

Get for last name

**Returns:**

String of student last name

### **getEmail**

```
public java.lang.String getEmail()
```

Get for student email

**Returns:**

String of student's email

### **getStudentNum**

```
public long getStudentNum()
```

Get for student number

**Returns:**

long of student number

### **setFName**

```
public void setFName(java.lang.String name)
```

Set for first name

**Parameters:**

name - String to be new first name

### **setLName**

```
public void setLName(java.lang.String name)
```

Set for last name

**Parameters:**

name - String to be new last name

### **setEmail**

```
public void setEmail(java.lang.String email)
```

Set for email

**Parameters:**

email - String to be new email

## **setStudentNum**

```
public void setStudentNum(long num)
```

Set for student number

### **Parameters:**

num - Long to be new number

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#) Detail: [Field](#) | [Constr](#) | [Method](#)

---

## Project Plan

| ID | Task Mode | Task Name                                  | Duration   | Start        | Finish       | Timeline |
|----|-----------|--|------------|--------------|--------------|----------|
| 1  | ✓         | Team Assignment 1                          | 13 days    | Tue 28/01/14 | Fri 14/02/14 |          |
| 2  | ✓         | Title Page                                 | 15 mins    | Tue 28/01/14 | Tue 28/01/14 |          |
| 3  | ✓         | Complete UML Class Diagram                 | 7 days     | Tue 04/02/14 | Wed 12/02/14 |          |
| 4  | ✓         | Complete User Stories with Pivotal Tracker | 7 days     | Tue 28/01/14 | Wed 05/02/14 |          |
| 5  | ✓         | Complete Personnel                         | 5 days     | Tue 04/02/14 | Mon 10/02/14 |          |
| 6  | ✓         | Install/Learn Microsoft                    | 2 hrs      | Tue 11/02/14 | Tue 11/02/14 |          |
| 7  | ✓         | Complete Project Plan                      | 1 hr       | Thu 13/02/14 | Thu 13/02/14 |          |
| 8  | ✓         | Milestone 1: Team Assignment 1             | 0 days     | Fri 14/02/14 | Fri 14/02/14 |          |
| 9  | ✓         | Team Assignment 2                          | 11.87 days | Fri 14/02/14 | Fri 28/02/14 |          |
| 10 | ✓         | Title Page                                 | 15 mins    | Tue 25/02/14 | Tue 25/02/14 |          |
| 11 | ✓         | Revised UML Class Diagram                  | 1 day      | Tue 25/02/14 | Tue 25/02/14 |          |
| 12 | ✓         | Revised User Story                         | 1 day      | Fri 28/02/14 | Fri 28/02/14 |          |
| 13 | ✓         | Completed Project Plan                     | 1 day      | Fri 28/02/14 | Fri 28/02/14 |          |
| 14 | ✓         | Prototype                                  | 4 days     | Tue 25/02/14 | Fri 28/02/14 |          |
| 15 | ✓         | Implement 6 User Stories                   | 4 days     | Tue 25/02/14 | Fri 28/02/14 |          |
| 16 | ✓         | Course Class                               | 3 days     | Tue 25/02/14 | Thu 27/02/14 |          |
| 17 | ✓         | Student Class                              | 2 days     | Tue 25/02/14 | Wed 26/02/14 |          |
| 18 | ✓         | Deliverable Class                          | 3 days     | Tue 25/02/14 | Thu 27/02/14 |          |
| 19 | ✓         | Grade Class                                | 1 hr       | Thu 27/02/14 | Thu 27/02/14 |          |
| 20 | ✓         | GUI and Implementation                     | 4 days     | Tue 25/02/14 | Fri 28/02/14 |          |
| 21 | ✓         | Serialization                              | 4 days     | Tue 25/02/14 | Fri 28/02/14 |          |
| 22 | ✓         | Milestone 2: Team Assignment 2             | 0 days     | Fri 28/02/14 | Fri 28/02/14 |          |