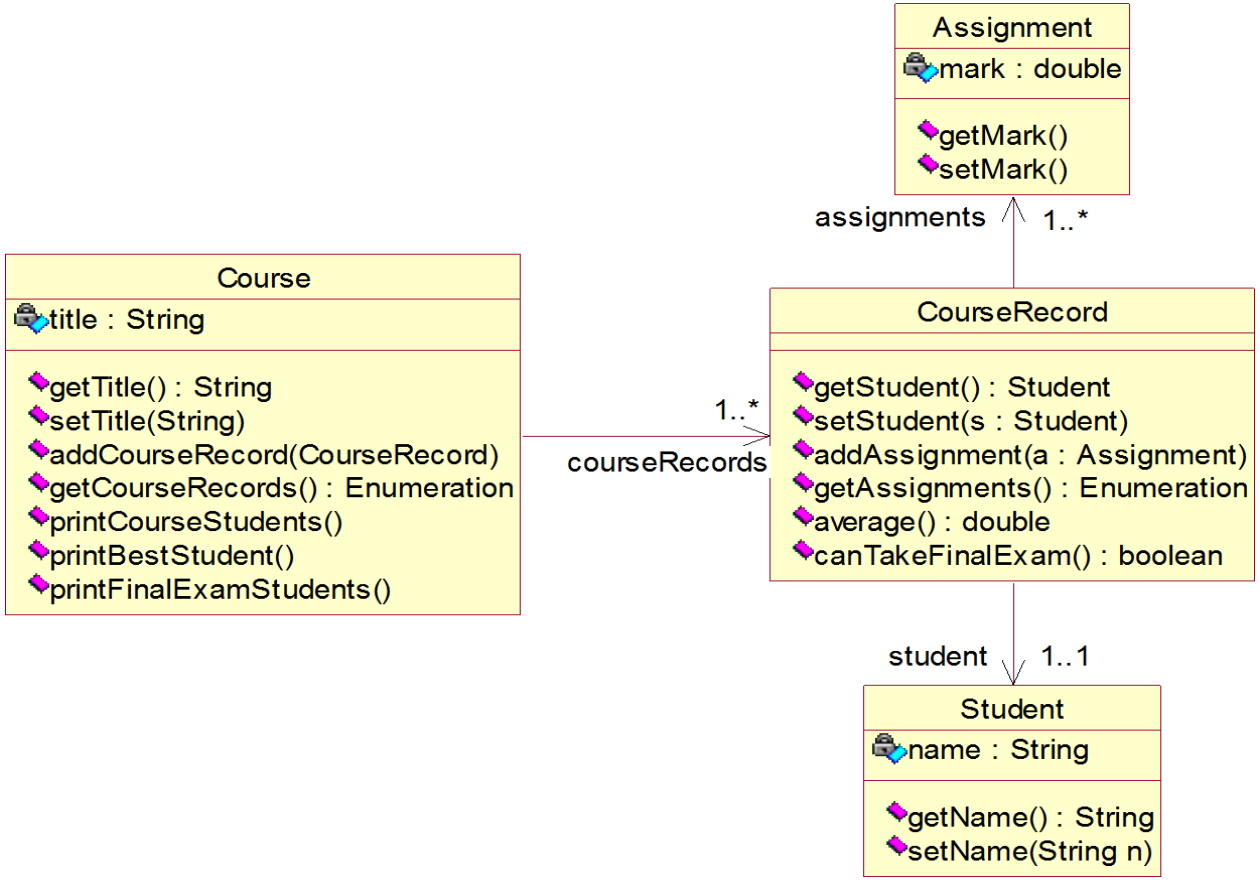
Assignment (1)
Java

Class diagram for the exercise



Student class

```
public class Student {
    private String name;

    Student() {
    }

    public String getName() {
       return name;
    }

   public void setName( String n) {
       name=n;
    }
}
```

Assignment class

```
public class Assignment {
    private double mark=-1;
    Assignment() {}
    public double getMark() throws NotYetSetException {
       if (mark==-1) {
             throw new
              NotYetSetException("Mark is not yet set");
         }
         return mark;
    }
   public void setMark( int m) {
       mark=m;
    }
}
```

CourseRecord class (1)

```
import java.util.Vector;
import java.util.Enumeration;
public class CourseRecord {
    private Student student;
    private Vector assignments;
    CourseRecord() {assignments=new Vector();}
    public Student getStudent() {
       return student;
    }
    public void setStudent(Student s) {
       student = s;
    }
  ...
```

CourseRecord class (2)

```
...
public void addAssignment(Assignment a) {
          assignments.addElement(a);
  }

  public Enumeration getAssignments() {
    return assignments.elements();
  }

...
```

NotYetSetException class

```
public class NotYetSetException
     extends Exception
{
    public NotYetSetException(String s) {
        super(s);
    }
}
```

Course class

```
import java.util.Vector;
import java.util.Enumeration;
public class Course {
    private String title;
    private Vector courseRecords;
    Course() {courseRecords=new Vector();}
    public String getTitle() {return title;}
    public void setTitle( String t) {title=t;}
    public void addCourseRecord(CourseRecord cr) {
       courseRecords.addElement(cr);
    }
    public Enumeration getCourseRecords() {
       return courseRecords.elements();
    }
  ...
```

Exercises for the CourseRecord Class

```
public double average() throws NotYetSetException {
    /*
     * Computes and returns the average for a student by
     * traversing the Vector of her assignments and
     * obtaining the mark for her assignment.
     * This method should handle the NotYetSetException that
     * can be thrown by getMark() by
     * ignoring the Assignment object that caused it in the
     * average computation
    */
    return 0.0;
}

public boolean canTakeFinalExam() {
    /*
     * Returns true if the student has a mark
     * for at least three assignments and false
     * otherwise.
    */
    return true;
}
```

Exercise for the **Course class**

```
public void printCourseStudents() {
    /*
     * Prints the names of the students
     * enrolled for this course
    */
}

public void printBestStudent() {
    /*
     * Prints the best student according to the
     * average computed by the CourseRecord class
     * for each student
    */
}

public void printFinalExamStudents() {
    /*
     * Prints the student names that can participate
     * in the final exam, based in the value returned
     * by the method "boolean canTakeFinalExam() of the
     * CourseRecord class
    */
}
```