# JAVA ASSESSMENT

Welcome to the end of course assessment for the Java Programming Fundamentals module!

In this session, you will complete the **StudentGen project.**

By the end of this session, you will be able to:

* Write the methods that complete the program.
* Test the program to verify if it works properly.
* Prepare the program to handle wrong data format insertion.
* Create Unit tests to automate the program validation.

It's time to see how much you learned about Java and Object Oriented Programming.

## **Part 1: Understanding the StudentGen project**

1. Download the source code and import the project using IntelliJ Idea or any other IDE you prefer.
2. Understand the project structure:

* Packages
* Classes
* Functionality

1. Run and test the project to get a deeper understanding of how it works (remember the persistence mindset!).

## 

## 

## 

## 

## 

## **Part 2: Implementing the Student and StudentService missing features**

1. Open the *Student* class (src/com/generation/model/Student.java) and implement the following methods:

public void enrollToCourse( Course course )

{

//TODO implement this method

}

public boolean isCourseApproved( String courseCode )

{

//TODO implement this method

return false;

}

public boolean isAttendingCourse( String courseCode )

{

//TODO implement this method

return false;

}

@Override

public List<Course> getApprovedCourses()

{

//TODO implement this method

return null;

}

1. Open the *StudentService* class (src/com/generation/service/StudentService.java) and implement the following methods:

public boolean isSubscribed( String studentId )

{

//TODO implement this method

return false;

}

public void showSummary()

{

//TODO implement

}

Hint: To show the summary use System.out.println() to print out to the console.

## **Part 3: Implementing the missing main method features**

1. Implement the method to *gradeStudent( StudentService studentService, Scanner scanner )* in src/com/generation/Main.java to have a fully functional program.
2. Test the program to verify it works as expected.
3. Create a new student.
4. Enroll the student to a few courses.
5. Grade the student.
6. Show the students and courses summary and verify that data is correct.

## **Part 4: Handling exceptions**

1. Register a new user providing a wrong date format.
2. Modify the createStudentMenu so it handles correctly the exception when a wrong date format is inserted by the user.
3. Catch the exception and show a proper message to the user.

## **Part 5: Writing Unit Tests**

1. Write 2 Unit tests for the class *StudentService*
2. Write 2 Unit tests for the class *CourseService*

## **Challenge Yourself**

1. Implement a way to store grades for each course a student is taking. There should be a way to update/set the score. Afterwards, fill in the public List<Course> findPassedCourses( Course course ) method in Student.java
2. Implement an additional feature in the menu options that will display the average grade of all the students subscribed to a given course.

## 

