# 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.
* Prepare the program to handle wrong data format insertion.

**\*Note**:

* Please create your own class(es)/ methods as deem fit, additional attributes/properties can be added in as well
* Assume that there is no duplicate courses enrolled per student

## **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!).

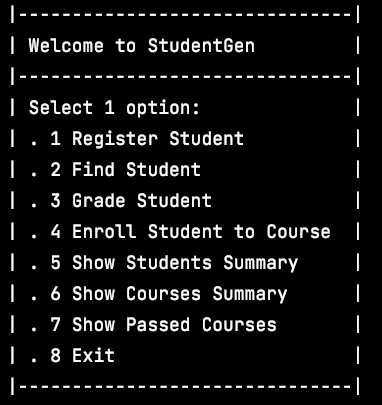
## 

## 

## 

## 

## **Part 2: Implementing the missing features for StudentGen Project**



## You are required to implement the missing features for the following options.

|  |
| --- |
| **Option (1): Register Student** |
| In ***StudentService*** class and implement the following method.  public void registerStudent( Student student )  {  //*TODO Add new student to the students hashmap*    } |

|  |
| --- |
| **Option (2): Find Student** |
| In ***StudentService*** class and implement the following method.  public Student findStudent( String studentId )  {  //*TODO Find the student from the Hashmap with the student id*  return null**;**  } |

| **Option (4): Enroll Student to Course** |
| --- |
| In ***StudentService*** class and implement the following method.  public void enrollToCourse( String studentId**,** Course course )  {  //*TODO check if students hashmap contains the studentsId, if have add the course to the student enrolledCourse*  }  In ***Student*** class and implement the following method.  public boolean enrollToCourse( Course course )  {  //*TODO Check if student has already enrolled to the course, if not add the course to enrolledCourses hashmap*    return false**;**  } |

| **Option (5): Show Student Summary** |
| --- |
| In the ***StudentService*** class and implement the following method.  public void showSummary()  {  //*TODO Loop through students hashmap and print out students' details including the enrolled courses*    }  In the ***Student*** class and implement the following method.  public HashMap<String**,** EnrolledCourse> getEnrolledCourses()  {  //*TODO return a Hashmap of all the enrolledCourses*  return null**;**  } |

|  |
| --- |
| **Option (3): Grade Student (Min grade input: 1 and Max grade input 6)**  In the ***Student*** class and implement the following method.  public Course findCourseById( String courseId )  {  //*TODO return the enrolled course with the course Id*  return null**;**  }  public void gradeCourse( String courseCode**,** double grade )  {  //*TODO set the grade for the enrolled Course*    }  In the ***StudentService*** class and implement the following method.  public HashMap<String**,** EnrolledCourse> enrolledCourses(Student student)  {  //*TODO return a HashMap of all the enrolledCourses*  return null**;**  }  public Course findEnrolledCourse( Student student**,** String courseId )  {  //*TODO return the course enrolled by the student from the course Id*  return null**;**  }  Open the **Main** class and implement the following method.  private static void gradeStudent( StudentService studentService**,** Scanner scanner )  {  Student student = *getStudentInformation*( studentService**,** scanner )**;**  System.*out*.println( "Enrolled course:" )**;**  //*TODO Loop through the student enrolled courses, and use the scanner object to get the course ID to insert*  // *the course grade*  } |

| **Option (7): Show Passed Course (the minimum passing grade is 3.0)**  **Min grade input: 1 and Max grade input 6** |
| --- |
| In the *Student* class and implement the following method.  public HashMap<String**,** EnrolledCourse> findPassedCourses()  {  //*TODO Check the enrolled courses grade and compare to the passing grade*  return null**;**  }  In the **Main** class and implement the following method.  private static void showPassedCourses(StudentService studentService**,** Scanner scanner )  {  //*TODO Loop through the student enrolled courses, and show all the passed courses*  } |

## 

## 

## 

## 

## 

## **Part 3: 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 and get user to enter a date again.

*Hint: In PrinterHelper Class, check for Date format: reference link:* <https://examples.javacodegeeks.com/core-java/text/parseexception/java-text-parseexception-how-to-solve-parseexception/>

|  |
| --- |
| In the **PrinterHelper** class and edit the following method.  public static Student createStudentMenu( Scanner scanner ) throws ParseException  {  System.*out*.println( "|-------------------------------------|" )**;**  System.*out*.println( "| . 1 Register Student |" )**;**  System.*out*.println( "|-------------------------------------|" )**;**  System.*out*.println( "| Enter student name: |" )**;**  String name = scanner.next()**;**  System.*out*.println( "| Enter student ID: |" )**;**  String id = scanner.next()**;**  System.*out*.println( "| Enter student email: |" )**;**  String email = scanner.next()**;**  System.*out*.println( "| Enter student birth date(MM/dd/yyyy)|" )**;**  DateFormat formatter = new SimpleDateFormat( "MM/dd/yyyy" )**;**  //*TODO validate date format and catch exception to avoid crash*  Date birthDate = formatter.parse( scanner.next() )**;**  System.*out*.println( "|-------------------------------------|" )**;**  Student student = new Student( id**,** name**,** email**,** birthDate )**;**  System.*out*.println( "Student Successfully Registered! " )**;**  System.*out*.println( student )**;**  return student**;**  } |