

Assignment #1

Due

Milestone (Submit Part 1): before 3 pm on Friday, September 14

Final Submission (Submit Parts 2&3): before 3pm on Friday, September 21

Part 1

Learning Outcomes: Upon successful completion of Assignment #1 Part 1 you will be able to:

- Edit, compile and execute a (single class) Java program
- Write Java programs that contain: expressions, input/output, arrays and static methods

Problem statement: Write a Java program that calculates and outputs a student's sessional Grade Point Average (GPA) for one term of courses, using the UVic standards.

Suggested Supporting Information to solve your problem:

- UVic Academic Calendar (<http://uvic.ca/calendar> search for 'grading')
- GPA worksheet (Found in Resources: Assignments: Assignment of the CSc 115 Connex Site. Note: this document was created for another activity, but may be instructive)

Specifications:

- The solution to the problem must use only one Java class, called **GPACalculator.java**.
- Your program must use two arrays, one to store the letter *Grades* that the student achieved on each course, and one to store the *Grade Point Value*
- Your program must use at least 3 static methods (in addition to the `main()` method.)

Submitting your Solution:

When complete submit your `GPACalculator.java` file to the CSc 115 Connex Site using the Assignments: Assignment 1 Milestone link before 3:30 on Friday, September 14.

This milestone is a formative exercise: It must be completed but will only be graded pass/fail, then used to inform the instructional team where pre-requisite learning needs support.

Part 2

Learning Outcomes: Upon successful completion of Assignment #1 Part 2 you will be able to:

- Use a Java class to instantiate objects and manipulate its attributes using the class methods

Problem statement: Using the `Student.java` class provided and the input file `studentData.txt` calculate and output a GPA for every student listed in the file.

The input file will have the following format: The first line will contain an integer indicating the number of student records contained in the file. For each student there will be: name, id number, followed by an integer indicating the number of terms the student has completed. For programming simplicity, we are assuming that every student completes 5 courses every term. Thus the number of terms integer will be followed by the appropriate number of terms.

Suggested Support to solve your problem:

- It would be a good idea to hand calculate the GPA's of at least some students in the file: Then you will have known correct results to use in the testing phase of your development. Also, it will assist you to design the algorithm.

Specifications:

- The solution to the problem must use be contained in only one Java file, called **`StudentGPAs.java`**.
- The **`Student.java`** class must be used, including using its methods, wherever possible but not altered in any way. Since you are not expected to alter the given file, you will **not** be handing in the `Student.java` file.

Submitting your Solution:

- When complete submit your `StudentGPAs.java` file to the CSc 115 Connex Site using the Assignments: Assignment 1 Submission link before 3:30 on Friday, September 21.

Part 3

Learning Outcomes: Upon successful completion of Assignment #1 Part 3 you will be able to:

- Create your own Java class.
- Test the functionality of the created class by instantiating objects. to instantiate objects and manipulate its attributes using the class methods

Problem statement: Write a java class, called `StudentPersonal.java`, that can be used by a student records system to maintain the personal information (as compared to the course and grade information of Part 2) of a University student. The personal information includes:

Name, Identification number, birthdate, Address (street), Address (city), address (province), address (country), address (postal code), current email, emergency contact (name), emergency contact (telephone number).

Include constructor(s), getter and setter methods, an equals method and a toString method. Create a test program that you can use to test the functionality of your class.

Suggested Support to solve your problem:

- The Student.java file that was used in Part 2 can serve as a solid guide.

Submitting your Solution:

- When complete submit your `StudentPersonal.java` file to the CSc 115 Connex Site using the Assignments: Assignment 1 Submission link before 3:30 on Friday, September 21.