This assignment will focus on top-down development. Testing and debugging methods will have to be used and explained during the discussion portion of the assignment review.

You will create an application for generating a letter grade based on a numeric grade entered for a given student. The interface must include a prompt asking for the numeric grade. The prompt should give the option of quitting the application. **After displaying the letter grade, the user should be prompted to enter another numeric grade for a different course for the same student or entering a new student record, or display the report or quit the application. On the report, you should be able to select students by grade achieved, by subject or a complete list of all students and in each case print the student name, course, grade achieved and letter grade. Bonus marks are available for those who enhance the deliverables further, including the interface.**

**Specifications**

The program prompts the user for a student name (first and last) along with a numeric grade in the range 0 to 100 or an option to end the entry of grades (Q). When a inputting student data, error checking logic should be used to ensure that data entered is a valid type (and range). Grades are distributed as follows:

Letter Grade Percent range

A+ 95-100

A 90-94

A- 85-89

B 80-84

C 70-79

D 60-69

F <60

**Sample Screen Layout**

Your screen must look like this or better, however the names below would represent the names a user types in. You also must make your program work for an infinite number of record entries:

**Welcome to Grade Converter**

**Please enter student First Name and Last Name: Jeff Snow**

**Please enter the percent achieved for Jeff Snow: 78**

**The letter grade for Jeff Snow is C.**

**Please press (I) to continue inputting records or press (Q) to quit entry: I**

**Please enter student First Name and Last Name: Jessica George**

**Please enter the percent achieved for Jessica George: 92**

**The letter grade for Jessica George is A**

**Please press (I) to continue inputting records or press (Q) to quit entry: Q**

***Thanks for using Grade Converter***

**Program Checklist**

|  |  |  |
| --- | --- | --- |
|  | **Description** | **Marks** |
| **Knowledge** | **Programming Concepts:**   * Appropriate declaration of variables with meaningful names and  suitable data types * Appropriate use of object classes for input and output * Appropriate use of arithmetic operators to perform program calculations * Appropriate use of methods * Appropriate use of arrays to store student information, courses, grades * Appropriate use of conditions & loops |  |
| **Thinking** | **Algorithms:**   * Flowchart provides detailed step-by-step instructions to properly implement the program specifications. |  |
| **Communication** | **Method documentation:**   * Uses pre and post method documentation   **Program Header:**   * contains programmer’s name, course code, date program written, program name and a comprehensive description of the purpose of the program   **Internal Documentation:**   * comments are used appropriately within the program and provide a meaningful summary of major processes   **Formatting:**   * program source code is properly indented where required and contains appropriate white space for readability * User interface is courteous, esthetically pleasing, and free of spelling and grammar errors |  |
| **Application** | **Implementation:**   * Output is formatted at minimum as displayed in the sample layout. * User input is formatted as specified * Program source code is efficient and executes as required with no syntax or logic errors * Error checking logic is used on all user input * **Bonus**: up to 1 additional bonus mark for improving the user interface beyond what was provided in the specification |  |