Lab 05 – 14-10-2024

Note 1: This is graded lab. Code in C++. Evaluation will be lenient but no compromise on cheating or any other violation. Therefore, please do your own work.

Note 2: Create following classes with constructors, setters (with checks for valid value and handle default value), getters per data member and required utility functions:

Task 1: The **Student** class has following attributes:

- name: The name of the student cannot be empty.
- age: The student's age (must be between 5 and 20 years).
- marks: The student's grade (must be between 0 and 100).

Constructors:

The constructor initializes the name, age, and grade. It uses setter functions to validate and assign these values. If invalid values are provided:

- The name defaults to "Unknown" if its length is less than 3.
- The name default to "NAME", if it is empty.
- The age defaults to 5 if it is not within the range [5, 20].
- The grade defaults to 50 if it is not within the range [0, 100].

Utility Functions:

- **void printStudentDetails()**: Prints the student's details (name, age, and marks).
- **bool isPassed()**: Returns true if the student's marks is 50 or more.
- **char getGrade ()**: Returns the grade category based on the student's marks:
  - 'A' for grades ≥ 85.
  - 'B' for grades ≥ 70 and < 85.
  - 'C' for grades ≥ 50 and < 70.
  - 'F' for grades < 50.

Task 2. The **Course** class has following attributes:

- courseName: The name of the course (cannot be empty).
- courseCode: A unique code for the course (must be a positive integer).
- credits: The number of credits (must be between 1 and 6).

Constructors:

The constructor initializes the courseName, courseCode, and credits:

If invalid values are provided:

- The courseName defaults to "COURSE" if it is empty.
- The courseCode defaults to 0 if it is not a positive integer.
- The credits default to 1 if they are not between 1 and 3.

Utility Functions:

- **printCourseDetails()**: Prints the course name, code, and credits.
- **bool isGreater(Course& c)**: Compare **the credits**

Task 3. **The** The **Library** class has a collection of books. It includes attributes:

- libraryName: The name of the library.
- books: A list (use vector<string>) of book titles.

Constructors:

The constructor initializes the libraryName and sets up an empty list of books:

Utility Functions:

- **addBook(string title)**:
  - Adds a Book to the library's book collection.
  - Takes a string parameter.
  - Example: library.addBook("Operating Systems");
- **removeBook(const string& title)**:
  - Removes a book by its title from the library's book collection.
- **printLibraryDetails()**:
  - Prints the library's name and the details of all books available in the library.
- **totalBooksCount()**:
  - Returns the total number of books in the library.