Practical project - Library

420-D02-SU

Introduction to structured programming

AEC Programming, Networks and Security (LEA.5F)

AEC Programming and Web Technologies (LEA.5G)

AEC Video Game Programming (LEA.CU)

**Evaluation weight:**

25% of the final grade

**Submission specifications:**

* **Due date: morning of the final exam**
* Name the program correctly: Library + *name1* + *name2* [ + *name3* ]
* The code of the program and any dependencies
* The compressed file deposited in Dropbox in the folder designated for this purpose

**Description:**

Create a prototype of a program that simulates a library. The program should randomly generate an abundance of books (Titles) and then store the books in it’s library. The user should then be able to search the library for the book they desire by giving either a part of the title or full title. The speed at which the program is capable of searching and its scalability are the primary focus.

The menu actions are the following

1. Add N books to library (User is prompted for int N, adds this many random books to library)
2. User can search the library
   1. User is prompted to input part of the title
   2. It displays all relevant search results, indexed
      1. User can enter more information to refine search further
      2. User can enter index to select the book
      3. If there are more than 10 results, the user is informed there are too many results to display, and is asked to refine the search further

**Remarks:**

* The books must be randomly generated, and the library should contain between 1 and 50000 books.
* All input must be validated.
* Each task out of the various functionalities should be delegated to an appropriate function.
* This project must be done in teams of two or three.

**Teamwork Validation**

This is a team project and must be worked together as a group, and not copied from any external sources.To verify that all members are familiar with the code, after submission each team member will be seperated and questioned about the project. The questions will be along the lines of “Show me the code that does X”, “What will this breakpoint reveal if I do…” or “If I change this code here, what will happen?”. If a member is found being unfamiliar with the code, their total mark will be multiplied by the amount they know. For example, if a member is shown to understand only 30% of the code, and the project received a 90%, their total mark will be 27%.

**Evaluation:**

The project will be graded according to the following criteria:

* **Exactitude of the program:**Does the program do what it is supposed to do?
* **Visual appearance:**All text displayed to the screen should be well arranged and written in proper English.
* **Input validation:**The program should properly handle errors that can occur during data input.
* **Structure of the code:**The program should be split up into functions according to the relevant needs.
* **Arrangement and clarity of the code:**The code should be properly indented, with relevant comments where needed, and it should respect programming conventions (variables in camelCase, constants in UPPER\_CASE\_SEPARATED\_BY\_UNDERSCORES). Hard-coded values should be put into constants (magic numbers).

|  |  |
| --- | --- |
| **Tasks to accomplish** | **Points** |
| Respect submission specifications | 1 |
| Program compiles and executes | 1 |
| Structure of the code (comments, etc) | 2 |
| Menu | 2 |
| Book generation | 3 |
| Name uniqueness | 3 |
| Book storage structure | 3 |
| Search algorithm | 4 |
| Display of relevant books | 2 |
| Exit the application | 1 |
| Input validations | 1 |
| Global evaluation (teamwork, effort) | 2 |
| **Total** (with bonus in parentheses) | 25 |