**Final Programming Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **Marking Guide:** | | | |
| ***Criteria*** | *Available Marks* | | *You mark* |
| Design | | /5 | |
| System menu and functions (except for simulation part) | | /4 | |
| Simulation & calculation (e.g., Safety rate, Damage & Survival rate), etc. | | /5 | |
| File I/O | | /4 | |
| Code Legibility (formatting, comments, etc.) | | /4 | |
| Program Structure (e.g., functions, data storage, efficiency) | | /4 | |
| Compiling | | /2 | |
| Executable and outputs | | /2 | |
| Bonus | | /5 | |
| Global variables, Breaking scope, etc. | | (-/10) | |
| **Total:** | | **/30** | |

**TASKS**

* Load names of up to 10 vehicles (Vehicle.txt file – store in array of structs), load names of up to 10 obstacles (Obstacle.txt file – store in array of structs)

\*\*\*\*\*\*reference vehicle structs from workshop3, include the 6 fields

**\*\*\*how to load from file to struct????????**

**MENU (allow user to navigate & select highlighted options)**

\*\*\*\*\*function for each item

\*\*\*\*\*use dummy function for whatever isn’t complete

**Search a vehicle**

* Receive user input to search for Vehicle\_name

\*search for info and print the vehicle information corresponding to input (**use if/switch statements???)**

\* display error if not found

**Search an obstacle (3 fields listed in doc, similar…)**

**Input/update vehicle info**

* Allows user to input info for a new vehicle and edit info for an existing vehicle

**Input/update vehicle info (similar….)**

**Simulate a vehicle crash against an obstacle**

* Prompt the user to select a vehicle and an obstacle

\*\*\*\*simulation details

* Prompt user to enter test conditions and save them to respective variables ()
* Simulate the crash scenario, test conditions (also inc velocity):
* Vehicle\_age(years, must not be greater than current date – manuf date or less than 0)
* Impact\_velocity(eter in kmph, calculate in m/s, must not be greater than top speed/less than 0)
* Number\_of\_passengers(must not be greater than # of seats in car)
* Number\_of\_seatbelt\_wearing\_passengers(must not be greater than # of passengers or less than 0)
* If **invalid data entered** print error + data<continue to prompt until valid value entered
* Calculations then print results and save data items as simulation results

CALCULATIONS

* Calculate impact\_force first
* Safety\_ rate
* Look at formula (based on impact\_force)
* Print the consitions and Safety\_rate
* Let user accept info or discard all conditions
* Damage\_rate
* Check formula details in sheet\*\*\*
* Survival\_rate
* Check more calc details
* If any passenger has no seatbelt 4 times less likely (how to know??**if statement seatbelt # > 1????**
* Any not wearing is 9 times less likely to survive than others

**Display simulation results**

* Table format (check sheet) \*\*\*\*
* Prompt user to input a vehicle number
* Display all test results for vehicle (**how to select all associated results from results.txt files???)**
* If vehicle chosen hasn’t been tested print message to tell user to test first
* \*\*check last conditions
* Offer option of saving this output to a file before quit/new simulation (array of structs, \*\*\*check sheet for details)

**Display all results**

* Display error message if no vehicle tested
* Print results for all vehicles if no error(everything from results.txt)

**Save vehicle/obstacle info**

* This option should extract all vehicle names from vehicle array and save back to Vehicle.txt, same for obstacles to Obstacle.txt for future use
* Only save up to 10 vehicles

**Save all results**

* Display an error if no vehicle tested (ie results.txt is empty/does not exist?)
* If no error create results.txt,save names and test results for all vehicles that have been tested