CST8333 Programming Language Research Project

# Practical Project Part 3 – Project Review 2 - See Blackboard for due date

* Refer to the Course Section Information (CSI) document posted in Blackboard under Course Information for additional requirements common to all assignments as well as details on the required use of the data set specified for the course.
* Refer to the documents in the dataset area in Brightspace for the list of dataset columns to use, note that all listed columns need to be used, and need to be present within your source code to verify you are using the dataset provided.
* The first record in the data set may contain the column names, if so you may skip over this record when reading in and parsing the data set data.
* You may use your previous work in this course as a reference / starting-point but I expect modifications, i.e. passing in older work again with none to very small changes will not earn marks.

## Tasks

* Augment your project, developed via Practical Project Part 2, by utilizing **one** advanced language feature from the list below, using the examples as a guide.
  + Inheritance and Polymorphism
    - Author your own super-class (or interface), and one or more sub-classes (implementing classes), then declare a method in the super-class (or interface) and override it in the sub-classes (implementing classes) and demonstrate polymorphic method calls.
    - E.g. Super class record object, sub-classed to change format of record output.
  + Database Connectivity
    - Create a database table with fields named after the column names, and use a database driver, to connect your program to a database. This can replace the reliance on the File-IO of the dataset. Your Create, Read, Update, and Delete options would then manipulate the database rather than the CSV file.
    - Populate the database table with all the data from the dataset, using field names in the database table
  + Data Structures (containers), Algorithms (that manipulate containers)
    - Use a List, Set, or Tree (etc.) rather than a simple array, and sort the records in the data structure based on single column from the dataset. You may use built-in API libraries to complete this task. Offer the user the option to use this functionality. (You must perform sorting using programming code (or an API call) and not an SQL statement sent into a database).
  + Generics
    - Create your own generic method or class and note that you need to define a generic class and then use it, simply using an existing generic class from the API is not enough.
    - You cannot use dynamically typed languages like Python to earn marks for this learning topic. To demonstrate learning you must actively write code using a generics feature. Python’s dynamic typing is, in a sense, generics done automatically i.e. there is no demonstrated learning and application of a generics topic.
  + Multithreading
    - Load the dataset from file using a separate thread of execution or use a thread (or asynchronous task) to enhance your program by doing something in parallel, e.g. use a thread of execution to display the records on screen or to update the data in memory.
  + Anything else you think might be interesting, check with your professor to confirm
    - Did you use a GUI? Then talk about the API
    - Did you use a web framework? (On-Rails, ASP.Net, ASP.Net MVC, Django etc.) Then talk about the API
    - Did you create a REST+JSON web-service? Then document how you used it.
  + Display your full name on screen so it remains visible at all times, or after each user interaction.
  + Provide the user the options and functionality to:
    - Reload the data from the dataset, replacing the in-memory data or use a database
    - Display all the records held in the simple data structure (for Practical Project Part 4 loading all records is required).
    - Create a new record and store it in the simple data structure or database
    - Select, display and edit a record held in the simple data structure or database
    - Delete a record from the simple data structure or database
  + Take a screen shot of your program performing the selected learning topic above, ensuring your full name is within the screen shot(s). (Not all learning topics will be visible on screen, for example the screen shot might not show multithreading visibly for example, but screen shot the program running the task that was updated to use multithreading)
  + Comment your source code files using documentation comments (docstrings in Python, XML-document in C# or VB.Net etc.)
  + Include at least one learning resource, for your learning topic, formatted using IEEE reference style.
* Write a single unit-test using a testing framework to test your new advanced topic.
  + Unit-Test Examples (you would only do one, or a similar test):
* If using a separate thread does it complete its task correctly?
* If using inheritance to format records with a parental record-type, and child formatted-record-types, is the overridden output format correct?
* If using a sorting algorithm, did the program sort the records correctly?
* Etc.
* Comment your unit-test source code using documentation comments (docstrings in Python, XML-document in C# or VB.Net etc.)
* Document your learning, demonstrate that your program runs and is unit tested with screen shots, and include the source code with your programmer comments within a single MS Word document. Also, include the original source code files as part of your submission. See the expected format, submission requirements, and grading guide below.

## Your single MS Word document should have this general format

* Cover page with your full name within it.
* Heading with name “Evidence of Learning”
  + Code figures to display the implementation of the Practical Project Part 3 language feature you selected.
  + Either:
    - Use small code examples for the topic, or
    - Indicate what line numbers in a larger code sample illustrate the concept.
    - **You must indicate clearly to your professor that you can identify what parts of your code illustrate and match to the programming concept.**
* Heading with name “Program Changes”
  + You need to outline where your advanced programming concept is located within your projects layered architecture. Alternatively, if you adopted a Framework (mobile / GUI / Web Site) or a design pattern like MVC, either on your own or as part of a framework, briefly outline how the framework structures your project. Discuss how the layered architecture or alternative architecture helps with refactoring your code to add features as well as assists with testing.
  + You are not required to provide any architecture or class diagrams here.
* Heading with name “Program Demonstration via Screen Shots”
  + Include screen shots of your running program; I should see records from the data set displayed, as well as creating, updating, deleting records and re-loading and saving records via user interaction with the program. Your full name must appear within the screen shots.
  + Indicate with brief description which screen shot shows the new feature in use, some features e.g. multithreading, database may have no visible presence on screen so use a brief statement indicating what is happening within the program e.g. “multithreading was used to asynchronously load the csv file at this point in program execution although the user interface was un-affected”
* Heading with name “Unit Testing Demonstration via Screen Shots”
  + Include screen shot(s) of your running unit test(s), this may either be within an IDE or run from a console. Your full name must appear within the screen shot(s).
  + Your unit test should test the new feature implemented where possible (GUI frameworks are an exception); minimally this must test your own code and not duplicate the same test from the previous Practical Project Part 2.
* Heading with name “Source Code Commenting Example”
  + Copy and paste all of the source code, including programmer comments, from one source code file from your project to demonstrate you can write **documentation-comments**. Use a font size of 10 point, with a monospaced font of your choosing.
  + The source code file you select must show the new advanced feature you worked on.
  + Copy and paste your unit test source code file, including programmer comments into this section. Use a font size of 10 point, with a monospaced font of your choosing.
  + Note: Some frameworks generate many code files, which you never edit yourself. Only include a source code file, and unit test file, you created or edited directly.
* Do not copy and paste code from the web into your demonstration program, it must be your own work. In other words, even properly cited and referenced code copied from a website will not earn marks, as you must provide your own work. Your professor while grading your submission will make this determination.

## Submission Requirements

* Upload your MS Word document and original source code files by the due date
  + One MS Word document
  + One zip file containing your original source code file(s)

\*\*Submitting any other format other than .doc or docx for your MS-Word document will result in zero for this assessment. Libra-Office users save-as… MS Word.

Do not bundle your MS Word document inside the source code zip-archive keep them separate instead.

* Ensure your full name is included in all materials as asked.

## Grading (Total 18 Points)

**Note: There is no entry in this rubric for a cover page with your full name in it, however a mark deduction of 3 points will be applied if you do not have a cover page in your MS Word document.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Poor/Missing (0) | Below Expectations (1) | Meets Expectations (2) | Exceeds Expectations (3) |
| Evidence of Learning | Poor/Missing or program does not use the dataset as required. Code samples are simple showcase items that do not integrate with the application, for example like cats and dogs for inheritance that have nothing to do with the program’s functionality. | New feature does not build and/or run or is incomplete. Code figures and / or descriptive text and / or figure-text missing for several items. Student provides hyperlink to learning resource. | New feature does build and/or run, but is not fully functional, i.e. does not work to produce correct results or does not meet expectations as specified. Code figures and / or descriptive text and / or figure-text missing for some items. Student provides reference, loosely follows IEEE reference style. | New feature is complete, functional, and working correctly meets or exceeds expectations specified. Code figures and / or descriptive text and / or figure-text present for all items. Student provides reference, closely follows IEEE reference style. |
| Program Changes | Poor / Missing or not separated out into file-based layers. E.g. A program consisting of a single code file with methods. Student does not outline where the advanced language feature is located. | Vague explanation of where the advanced language feature occurs in the layers, or vague overview of an alternative framework driven architecture. | Student discusses where the advanced feature occurs within the layers with detail. Alternatively, the student discusses the new framework architecture. | Student discusses how the layered architecture, or alternative architecture, has helped with refactoring and testing of new program features. |
| Screen Shots, Running Program | Poor/Missing  E.g. missing full name from all images, image file not within the MS Word document. Screen shot does not demonstrate use of the dataset. Screen shot does not show or indicate selected advanced language feature. | Screen shots are within MS Word document. No explanation of the image. Has partial name or nickname or but full name missing from some screen shots. Some project functionality missing or program crashes. The advanced language feature is partly complete. | Screen shots are within MS Word document. Student provides brief generalized description of each image. Has full name in nearly all screen shots. Most of the requested project functionality is working. Descriptive text adjacent to screen shots briefly indicates presence of advanced language feature. | Screen shots are within MS Word document. Student provides brief yet detailed description of each image. Has full name in nearly all screen shots. All project functionality is documented and working. Descriptive text adjacent to screen shots indicates and discusses advanced language feature. |
| Screen Shots, Running Unit Test(s) | Poor/Missing  E.g. missing full name within images, image file not within the MS Word document. Screen shot does not demonstrate unit testing with a framework. | Screen shots are within MS Word document. No explanation of the image. Has partial name or nickname but full name missing from some screen shots. Test framework runs but does not actually test anything meaningful or crashes, does not test advanced language feature. | Screen shots are within MS Word document. Student provides brief generalized description of each image. Has full name in nearly all screen shots. Test framework runs and performs a basic or rudimentary test on advanced language feature. | Screen shots are within MS Word document. Student provides brief yet detailed description of each image. Has full name in nearly all screen shots. Test framework runs test and tests aspect of program related to advanced language feature selected. |
| Source Code, programmer comments. | Poor/Missing  E.g. missing full name as programmer comment at the top of the file as author of the file. | Student uses minimal comments in source code, e.g. the student does not comment (m)any class members. | Student comments some class and class members, however does not use documentation comments. | Student uses documentation comments in an accepted coding style specific to their language of study. If the language does not support documentation comments student provides evidence of this from reputable source, yet still comments code following best practices. |
| Source Code and Source Code files | Poor / Missing or MS Word document is bundled inside the code zip archive. | Not used for this criteria | Not used for this criteria | Student uploads both MS Word document, and source-code zip file keeping them separate, i.e. MS Word document is not inside zip file. |

## Additional Notes

**Video Game Software projects are not acceptable in this course.**

Your source code within the MS Word document, should match the code in your source code files, this includes the programmer comments. If there are large or many differences, you will lose marks.

**If you do not submit your full source code in a zip archive to so your code figures can be verified this entire assessment will be awarded zero points. It is your responsibility to submit the correct files.**

Name your project something similar to CST8333ProjectByYourName, where ‘Your Name’ is your ACSIS name. Where you develop the project through Practical Project Parts 1 through 4, naming your project “Project 1” (etc.) will not look professional for future submissions.