CST8333 Programming Language Research Project

# Research Assignment 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 assessments.**

## Tasks

Complete each of the tasks below, you may use bullet style rather than essay style for your MS Word document.

The tasks titles themselves can be used as headings in your MS Word document.

**Caution: Many students will find Practical Project Part 2 to be the most challenging activity in the course, budget your time here in this activity quickly (be brief), and then start your research, learning, and start working on Practical Project Part 2 early.**

1. Development Environment Changes

* If your development environment has changed because of your learning, document it here with version numbers, otherwise report that there are no changes to the development environment.
* Large changes, like a different programming language, can be done at this point in the course before starting Practical Project Part 2. Moderate changes, like changing project type between web framework, desktop GUI, mobile, or console can also be done at this point in the course. **After Practical Project Part 2 is completed, it will be very difficult to make changes to programming language or project type. Consult your professor promptly for guidance if you have any doubts.**
* **Note: You may realize that your project may be too ambitious (advanced frameworks) for you to complete the coursework as you proceed with Practical Project Part 2; you are permitted to reduce your project to a console-based program and / or change to a different programming language than you used earlier while still meeting the requirements for Practical Project Part 2.**

1. Research and Learning Resources
   * Research locating learning resources and examples on N-Layered and/or Model View Controller (MVC) architectures to determine what architecture you will use in your Practical Project Part 2 project. Also, see reference examples provided in Brightspace, which were created in Java.
2. WBS and Gantt Chart for Practical Project Part 2

* Review the handout provided with WBS and Gantt Requirements, Examples, and Resources.
* Review the handout for Practical Project Part 2.
* Create a brief Work Breakdown Structure (WBS) to decompose the Practical Project Part 2 tasks into smaller sub-tasks.
* Create a Gantt Chart in MS Project or Project Libra based on your WBS;
  + Indent to create sub-tasks;
  + Link sequential tasks;
  + Create one milestone for “Practical Project Part 2 Delivered”, set with the due date
  + Add your full name in the Resource column adjacent to at least one task or sub-task.
* Insert a screen shot of your Gantt chart into this section of the MS Word document.
* Submit the Original Gantt Chart file, along side your MS Word document.

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

* Cover Page
* Headings as above with content addressing the questions and tasks including screen shots where applicable.
* References used in your research and write up.

## Submission Requirements

* Upload your single MS Word document and your Gantt chart.
* Submitting any other format other than .doc or docx for your MS-Word document will result in zero for this assignment. Open-Office/Libra-Office users save-as… MS Word.
* Your Gantt chart must be submitted as either Microsoft Project or ProjectLibre with file extension .mpp or .pod (respectively), any other file format will score zero for this part. If the original file is missing, i.e. there is a screen shot only, you may loose marks.
* Ensure your full name is included in all materials as asked; you will lose marks if your full name is not included on a cover page as the first page of the document.

## Grading (8 Points Total)

**Note: A mark deduction of 3 points will be applied if you do not have a cover page with your name in it.**

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Missing / Poorly done (0) | Below Expectations (1) | Meets Expectations (2) |
| Development Environment | Missing or done very poorly or wrong file type. | Does not meet all of the requested requirements: If there are changes the new operating system, tools, and language, (etc.) student lists and documents with version numbers. Alternatively, student provides a statement that there are no changes to the development environment. | Meets all of the requested requirements: If there are changes the new operating system, tools, and language, (etc.) student lists and documents with version numbers. Alternatively, student provides a statement that there are no changes to the development environment. |
| Research and Learning Resources | Missing or done very poorly or wrong file type. | Student lists learning resources related to the specified architectures however does not use IEEE reference style. | Student lists learning resources related to the specified architectures, and uses IEEE reference style. |
| WBS | Missing or done very poorly or wrong file type. | Does not meet all of the requested requirements: breaks down the tasks from Practical Project Part 2 into sub-tasks, no dates or times are used, sub-tasks are indented, has expected numbering format. | Meets all of the requested requirements: breaks down the tasks from Practical Project Part 2 into sub-tasks, no dates or times are used, sub-tasks are indented, has expected numbering format. |
| Gantt | Missing or done very poorly or wrong file type. | Does not meet all of the requested requirements: original Gantt chart file provided, tasks and sub-tasks match WBS, sub-tasks indented, time estimates in days, sequential tasks linked, one milestone, and student name as resource. | Meets all of the requested requirements: original Gantt chart file provided, tasks and sub-tasks match WBS, sub-tasks indented, time estimates in days, sequential tasks linked, one milestone, and student name as resource. |

Writing a paper on a non-permitted language (see below), e.g. Java will result in a score of zero.

## Additional Notes

### Programming languages recommended / permitted for study

* Python: Desktop App either console or GUI with Tkinter, or Web with Django
* C#: Desktop App with WPF or Web with ASP.Net MVC
* C: This looks similar to Java, but it is more difficult to program in, recommended console program
* C++: See notes on C above, avoid C++ .Net, and use C++ Standard Edition console program.
* Ruby on Rails: Linux recommended, get a virtual machine if using Windows.
* Swift / Objective-C: Only select if you already have an Apple computer (Macbook air/pro) to run Xcode legally.
* Kotlin: Android development
* Server-Side JavaScript, e.g. node.js
  + You may use a client-side JavaScript framework in addition to a server-side JavaScript framework to create a client, or you may use a testing tool like Postman instead for testing instead of a client.

### Programming languages not permitted for study.

* Java
* COBOL
* C++.**Net** (This is a Microsoft Extension to C++, not well supported, and problematic for learning C++)
* Android with Java
* HTML, PHP, client-side JavaScript
* Declarative / Functional languages e.g. Clojure, Prolog, Lisp etc.
* **Video game software projects are not permitted for this course.**

If you would like to study a programming language not listed above, contact your course professor first to get a determination and written (emailed) permission to proceed.