

## Capstone Course Project

### Guidelines, Deliverables & Rubric

### 1200 Points

The Capstone course is the culmination of the student's academic and professional experience in the Master of Quantitative Finance with Focus on Alpha Generating program. Over the duration of this course, students will apply their knowledge gained during the program, integrating the skills necessary for analyzing issues, thinking creatively, working collaboratively, and presenting impactful alpha generation strategies to create a Capstone Project. The Capstone Project will leverage all aspects of Quantitative Finance taught in the course of the program – technical, fundamental, behavioral etc and hopefully should be one of the most aspirational pieces of work a student completes in his or her career.

## Course Requirements

The final grade for the Capstone project would be from 1200 points and will be accounted as follows –

- A. **Project Proposal** - 200 points
- B. **Status Report** - 50 points;
- C. **Capstone Project** - 500 points;
- D. **Final Report** – 200 points
- E. **Project Presentation** - 200 points;
- F. **Peer review of other Capstone Projects/Papers** – 50 points

## Group Participation

To determine individual grades in a group assignment like the Capstone Project, team members will prepare one document that explains on what portion each team member worked, individual percentage of time spent, and estimated hours individuals dedicated to the two status papers, presentations and capstone paper. The document should be a collaborative effort, about 1-2 pages in length and due by the final day of submission. Factors to consider include level of effort for research, writing, editing, document creation, team coordination, contribution to discussion, participation and attendance in group meetings. The group participation document would not be graded itself and hence would not count towards the overall grade of the team.

## Grading Scale

The course would follow a relative grading structure. The mean and standard deviation of the overall performance of teams would be used to come up with a grading scale where projects are rewarded/penalized based on their deviation from the mean results of the all the teams. The exact dynamics of the grade allotment is prerogative of the professor.

The Rubrics to be followed for marking the individual segments of the course requirements are explained below.

## A. Project Proposal (200 points)

	Below Expectation 0-40 Pts	Meets Expectations 41-80 Pt	Exceeds Expectations 81-100 Pts
<b>Does the project Proposal Document clearly delineate the project idea</b>	No content is presented or the content is not relevant	Incomplete project proposal and missing part of required content. Goal of project and its relevance in Quantitative Finance applications not clear.	Project Proposal is clear and complete. Includes a clear project description, explaining the importance of the project, team qualification in regards to the project, what courses or knowledge you will apply, Team Member roles, main tasks and schedule, any expected issues or obstacles, any dependencies
<b>Does the Project Proposal clearly identify the goals and its own relevance in modern quantitative finance</b>	Project Objective is missing, unclear, or incomplete.	The project proposal talks about the objectives of the project and provides some idea as to its relevance.	Project proposal clearly identifies pain-points in current quantitative finance strategies, points out how the current project can help in the context and also discusses other possible real life applications of their research/project idea.  Overall clear and concise.

## B. Interim Status Report (50 points)

An interim status report is expected to be submitted by the students' midway through the duration of the Capstone Project

	Below Expectation	Meets Expectations	Exceeds Expectations
	0-20 Pts	21-40 Pt	41-50 Pts
<b>Relevance and Clarity of the Interim Status report</b>	No content is presented or the content is not relevant	Incomplete details of scope of the project already covered, possible roadblocks, changes in initial project objectives.	Clear and concise details of project scope and area already covered, update on project objectives etc.

## C. Capstone Project (500 points)

### Guidelines

Students can either use **Python** or **WebSim®** for the Capstone Project depending on suitability. **WebSim®** can be an optimal fit for projects focusing on trading strategy development while **Python** is an all weather language that can be leveraged everywhere. We leave it to the students to decide what works best for them.

In either case, the submitted code should constitute a fully workable version that can be easily validated and tested. In case of Python projects, all relevant module import statement should be part of the code. Students are encouraged to avoid usage of any special Python packages for the project and stick to using standard Python libraries mentioned as part of the course. In case such a non standard package is anyway used, students should provide clear directions as to how to access and install the same. Pip installations are preferred. In case a database engine is required to be used, please use MySQL only and submit the database schema files for validation.

### Capstone Project Deliverables for Technical Implementations

**Solution Design:** Write a 1-2 page document that lists the steps you would need to take to build your project. Please ensure that the document is understandable by someone with no programming background. If relevant, also include class diagrams that highlight the classes and the objects you intend to create in your program.

As you work on your code, you may find that your original solution design needs to be revised. That is perfectly fine. Redesigning your solution as you run into roadblocks or find better approaches is part of the process. In this case, please update your solution design document to reflect any changes and remember to provide details of any significant changes in approach during the interim status report submission phase. The solution design you submit should be the one implemented in your code.

**Code:** All your source code, i.e., the Python programming files or link to your shared WebSim Alpha generation programs.

**README File:** Provide all necessary instructions that will enable an end-user to install the required files and run your project.

**For Python Implementations**

- ✓ What version of Python is your project written in? If your program needs to be run in Python 3, specify this in the README. If you have used Python 2.7, specify this in your README.
- ✓ Your code should be able to be run on any platform. In rare/unavoidable cases if your code will run only on Windows, only on Mac, etc., the README should specify this. For safety's sake, even if you think the code is platform-agnostic, include details regarding the operating system and version you used to develop your project.
- ✓ If you used any external libraries, provide detailed instructions on how to download and install the files:
  - If you have used a package manager such as pip, please provide the exact command to be typed for downloading and installing the library.
  - If you downloaded the file manually, provide the URL to the download site.
  - If there are several versions of the external library, be sure to specify which version is needed.
  - If there are any known "pain points" that the user should keep in mind while installing or running the external library, be sure to include instructions or URLs regarding those.

Also explain how your program is to be run. Ideally, a user should be able to launch it from the command-line interpreter. If your code needs a particular environment, please specify this. But, please note that the use of any particular esoteric environment is strongly discouraged and you should ideally be able to deliver files that can be run on any platform without need of specific environments.

**For WebSim® Implementations**

- ✓ Basic details of your WebSim user account, the strategy name, the data sources use, how to interpret results etc

The README file need not be very long, but it should provide all the necessary details that will enable an end-user to set up and run your project as you have designed it.

**Analysis of Results:** Write a 2-page document (2000 words or fewer) analyzing the results obtained and drawing inferences and insights about the applicability of these results on actual live trading system development for alpha generation across different world markets. Do a proper S.W.A.T analysis and highlight the most important points.

## Rubric for Capstone Project (500 points)

This rubric is here to help you understand the specifications for how your project will be evaluated. It is the same rubric that the person evaluating your project will use. You should look at the rubric before you begin working on this project and before you submit it.

	Below Expectation 0-25 Pts	Meets Expectations 26-50 Pt	Exceeds Expectations 51-70 Pts
<b>Does the Solution Design Document clearly outline the steps needed to solve the problem</b>	<p>The solution design is vague or not provided, or does not list the steps to solve the problem.</p> <p>The solution design cannot be understood without a programming background</p>	<p>The steps needed to solve the problem are clearly listed.</p> <p>The design includes pseudo-code that can be read and interpreted by people with no programming background.</p>	<p>Brilliant solution design document, clearly outlined and easy to read.</p> <p>Addresses current project and suggests small extension of the work into related real life quantitative finance scenarios</p>
<b>Is the README file present and complete, clearly providing all the necessary instructions for running the program?</b>	<p>The README file is missing, unclear, or incomplete.</p>	<p>The README file provides full and clear instructions regarding program setup.</p>	<p>Industry standard README file, at par with the ones available with standard desktop software.</p>
<b>Does the code fully and satisfactorily solve the problem or implement the idea listed in the problem definition?</b>	<p>The code does not work in accordance with the ideas in the problem definition.</p>	<p>The code meets all the requirements listed in the problem definition.</p>	<p>Code that not only meets requirements but also implements the said functionality in the most optimal way.</p> <p>Code designed with live implementation in mind and structured in a way that is amicable to easy expansion of functional footprint.</p>
<b>Does the code run without errors?</b>	<p>The code crashes during execution.</p> <p>Required inputs to the program are not validated before use.</p>	<p>The code runs satisfactorily and without errors.</p> <p>If the program requires user input, all input is appropriately validated before use.</p>	<p>Extensive error handling for unforeseen exceptions.</p> <p>Implementation of configurable logging framework for back end validation.</p>
<b>Is the program intuitive to use?</b>	<p>The user might be confused and not understand what is expected during program</p>	<p>The user can run the program straightforwardly, without finding it confusing</p>	<p>The user interface is intuitive and polished.</p>

	<p>execution.</p> <p>User input is required, but the user cannot easily follow what is needed.</p> <p>If user input is required, the program does not print an appropriate message when the input is invalid.</p> <p>Reporting output is difficult to understand and corroborate to program.</p>	<p>or too complex.</p> <p>If input is required, the program clearly indicates to the user what input is expected.</p> <p>If input is required, the program prints user friendly error messages when the input is invalid.</p> <p>Neatly structures tabular output from program. Clearly labeled and annotated graphs wherever necessary</p>	<p>Provides end user the ability to vary input parameters multiple times to verifying program output without need to re-execute the program from beginning.</p> <p>Intuitive dashboards for showing program output supported by interactive and publication quality graphs</p>
<b>Are comments used effectively to explain different parts of the code?</b>	<p>Code is not easily readable or is not commented.</p>	<p>Functions, classes, and modules have adequate comments such that a fellow programmer can understand the purpose of the program.</p>	<p>Commenting structure that not only describes coding logic but also the functional logic behind the code. Clearly refers to URLs / libraries / other relevant content.</p>
	<b>Below Expectation</b>	<b>Meets Expectations</b>	<b>Exceeds Expectations</b>
	<b>0-30 Pts</b>	<b>31-60 Pt</b>	<b>61-80 Pts</b>
<p><b>For Python Implementations:*</b></p> <p><b>Are proper Python programming design guidelines followed?</b></p>	<p>Similar coding constructs (classes, functions, variables, etc.) do not use similar naming conventions (CamelCase, snake_case, ALL_CAPS, etc.).</p> <p>The names for variables, classes, and procedures are inconsistent and/or not meaningful.</p>	<p>Similar coding constructs (classes, functions, variables, etc.) use similar naming conventions (CamelCase, snake_case, ALL_CAPS, etc.).</p> <p>The naming scheme for variables, classes, and procedures is consistent with the <a href="#">PEP 8 Style Guide</a>.</p>	<p>Complete adherence to "Pythonic" idioms. Every line of code should be "Pythonic" and follow the common guidelines and express its functional and technical intent in the most readable way.</p> <p>Refer to <a href="http://docs.Python-guide.org/en/latest/writing/style/">http://docs.Python-guide.org/en/latest/writing/style/</a> for fully Pythonic code structure</p>
<p><b>For WebSim® Implementations:*</b></p> <p><b>Are proper WebSim® programming design guidelines followed?</b></p>	<p>Fails to clearly map the project idea to a alpha expression in WebSim®.</p> <p>Analysis steps of simulation results not clearly delineated.</p> <p>No fine tuning of</p>	<p>The projects starts with implementation of a relevant idea, maps it to an alpha expression, improves it, analyzes simulation results and fine tunes performance using different simulation settings.</p>	<p>The project not only maps the relevant idea to a proper alpha expression, but suggests different approaches and related improvements.</p> <p>Proper analysis and fine-tuning steps clearly implemented through</p>

	performance achieved		different simulation settings for all suggested Alpha expressions.

\*Depending on whether the project has been implemented in Python or WebSim®, only one of the relevant sections of the Rubric would be considered

## D. Final Report (200 points)

Students are expected to submit a 5-6 page project report on completion of their technical project work. This report should clearly analyze and explain the results, provide insights on the findings from the point of view of Alpha Generation and explain in detail about the usefulness and applicability of the findings in real quant trading scenarios. All papers will be archived in the WQU system and available for possible public viewing, and thus all papers must meet APA style guidelines. For more details on the APA style refer to the Tutorial at <http://www.apastyle.org/>

	Below Expectation	Meets Expectations	Exceeds Expectations
	0-20 Pts	21-40 Pt	41-50 Pts
<b>Relevance and Clarity of the Capstone Report</b>	No content is presented or the content is not relevant	Incomplete details of scope of the project already covered, possible roadblocks, changes in initial project objectives.	Clear and concise details of project scope and area already covered, update on project objectives etc.
<b>Functional Understanding</b>	The project is executed as a pure programming exercise.  The project design document, comments, structure, user input prompts, output formatting etc shows neither understanding nor appreciation of the real life Quantitative Finance problem that is being addressed	The project is executed as an applied programming exercise.  The project design document, comments, structure, user input prompts, output formatting etc shows clear understanding and appreciation of the real life Quantitative Finance problem that is being addressed	The project correctly identifies the appropriate Quantitative Finance context and extends application onto related real-life scenarios.
	Below Expectation	Meets Expectations	Exceeds Expectations
	0-30 Pts	31-60 Pt	61-100 Pts
<b>Analysis of Results</b>	Little time spent on	Good response, well thought out, thoroughly	Well articulated response that shows a thorough

	<p>composition and response.</p> <p>No real analysis done or conclusions drawn are vague and irrelevant. Does not add enough to the existing knowledge base and overall limited in scope.</p> <p>Sketchy or irrelevant S.W.A.T analysis.</p>	<p>composed.</p> <p>Applicability of project tasks in real life scenarios is clearly understood and articulated. The insights drawn are in-line with and significant enhancements to the alpha-generation ideas shared throughout the course.</p> <p>Detailed and insightful S.W.A.T analysis</p>	<p>understanding of the scenario while providing relevant academic citations references to real events from financial markets</p> <p>Provides novel ideas about the applicability of the results of the project in real life trading system design and application</p> <p>Brilliant S.W.A.T analysis, that not only correctly identifies pain points and opportunities, but also suggests innovative ways to control weaknesses and leverage strengths</p>

## E. Project Presentation (200 points)

	Below Expectation 0-15 Pts	Meets Expectations 16-30 Pt	Exceeds Expectations 31-40 Pts
<b>Enthusiasm</b>	Shows some negativity toward topic presented	Occasionally shows positive feelings about topic	Demonstrates a strong, positive feeling about topic during entire presentation
<b>Speaking Skills</b>	Presenter's voice is low. The pace is much too rapid/slow. Audience members have difficulty hearing presentation	Presenter's voice is clear. The pace is a little slow or fast at times. Most audience members can hear presentation	Uses a clear voice and speaks at a good pace so audience members can hear presentation. Does not read off slides.
<b>Subject Knowledge</b>	There is a great deal of information that is not clearly integrated or connected to the research.	Sufficient information with many good points made, uneven balance and little consistency	An abundance of material clearly related to the research is presented. Points are clearly made and evidence is used to support claim
<b>Organization</b>	Audience has difficulty following presentation because the presentation jumps around and lacks clear transitions	Information is presented in logical sequence which audience can follow	Information is presented in a logical and interesting sequence which audience can follow. Flows well
<b>Visuals</b>	Visuals are used but not explained or put in context	Appropriate visuals are used and explained by the speaker	Excellent visuals that are tied into the overall story of the research



--	--	--	--

### F. Peer Review of Other Capstone Projects (50 points)

As part of a peer review process, every team would be expected to critically examine the Capstone project submissions (Proposal, Project, Status Report, Final Report, Presentation) of another group using the same rubric as provided in this document. Based on their analysis, every group should submit a report containing their analysis clearly mentioning the scores, strengths, weaknesses and opportunities missed (things the project could have addresses but did not)

	Below Expectation	Meets Expectations	Exceeds Expectations
	0-20 Pts	21-40 Pt	41-50 Pts
Relevance and Clarity of the Peer Review Report	No content is presented or the content is not relevant	Incomplete review, or demonstrates an inaccurate judgment of others’ work	Complete and demonstrate a sound judgment for other group projects.