Team Assignment 02:

This second team assignment is the first part of your final project and will count toward your final project grade. It is a combined writing and a programming assignment. It must be completed in teams of 3 students enrolled for credit, unless otherwise approved in advance. Your team will be required to submit 1) a written proposal document and 2) your team's git source code repository. The proposal outlines your detailed plan for completing the final project. The code submitted shows initial feasibility of software development using the libraries you specify. All team members are expected to contribute equally to the submitted document and developed code. As with any group project, the key to ensuring a successful outcome is to start early and meet often.

1 Assignment Description

Your supervisor liked the market research report that your team submitted and now wants to pursue expansion into this application area of scientific computing. Your team has identified a specific problem defined within this application area to break into the market. Since you work for a fast-paced research and development group in Silicon Valley, funding is too tight to carve out any seed money for new endeavors. Your team will have to write a short proposal to acquire these seed funds from a highly competitive venture capital firm. An important aspect of the written proposal is showing some exciting preliminary results of your investigation into solving the problem you define. This proposal shouldn't be too difficult to write given your solid track record and extensive on-the-job experience with C++ software development.

2 Application Areas

You should choose one of the following major application areas for the final project:

- Convolutional Neural Networks
- Data Visualization
- Embedded Electronics
- Finite Element Modeling
- Graphical Processing Unit (GPU) Acceleration
- High Performance Computing
- Application Development
- Multi-Threading and Sockets

Within these application areas, you must define a specific problem as your primary focus. It is highly encouraged that you run ideas by me before beginning the assignment. If you cannot attend the office hours, you should communicate via email or setup an alternative time to meet with me as a team. I am happy to provide feedback on any problem ideas your team has, or to clarify the scope of the work I would expect to see.

3 Written Proposal

Your written proposal should provide the evaluators with sufficient justification for moving forward with the software development project. It should at a minimum:

- have a compelling and descriptive title,
- clearly define the problem to be solved,
- highlight the overall significance of the opportunity,
- describe your solution by breaking it down into goals, objectives, and (assignable) individual tasks,
- provide adequate technical details about any related projects, and
- include a detailed schedule with weekly milestones to which you will be held accountable.

Your document should be approximately 15-20 pages in length with figures. The main body text should use 10 point font and single spaced paragraphs. The organization should conform to the following standard outline:

3.1 Cover Page

- Include the project title and optional subtitle, preferably in a large, bold font.

- Proposal authors should be listed in alphabetical order by last name.
- The date of proposal submission and the date of proposed completion should be prominently shown, as well as the duration of the effort (in weeks).
- Include a logo of your company (Brown U.), the physical company address, and email address of the proposing individuals.

3.2 Table of Contents

List the major sections and subsections of the proposal with page numbers.

3.3 Glossary of Terms (optional)

Include a glossary if you make use of numerous specialized terms, mathematical variables, or acronyms.

3.4 Brief Statement of Work

This section should outline at a high level the scope of the problem, goals and objectives of the effort, and your basic technical approach. What are you trying to accomplish in the proposed final project? Limit this section to a half-page.

3.5 Significance of the Opportunity

Explain why this work should be done. What is the importance of completing this work? How will it be useful to the company when completed? Why should the sponsor fund this software development proposal?

3.6 Technical Objectives

State your technical objectives that provide a solution to the problem being addressed. Describe how your software will provide a new capability that does not exist. Provide a short paragraph explaining how each technical objective will satisfy the overall goal(s).

3.7 Technical Challenges

What difficulties do you envision in executing this proposed work? Identify a few major technical issues that you are aware of that may directly impact schedule or performance of the software development. Major technical issues should be identified if they pose a risk to on-time completion of your development effort. State how you will mitigate these risks or work around them with an alternative path.

3.8 Technical Approach

This section organizes your proposed work into separate, severable tasks, each related to one or more of the objectives you identified. Individuals responsible for each task (even if joint) should be identified in the task name. The tasks can be performed sequentially or simultaneously, but the duration of effort should be estimated. Include any significant milestones in your software development. Remember, milestones should be significant and measurable. Include a software test or some other objective metric to validate that your milestones were completed.

3.9 Proposed Software Architecture

The software architecture should be well thought out and described in this section. You should spend considerable time organizing your software project with your team such that there is a logical flow. Include block diagrams to show the program flow and a graphical representation of and any interaction between the various components or libraries of your software. Explain and show how you will include concurrency in your software (refer to Section 4).

3.10 Project Schedule and Milestones

You will be required to show clear objective evidence that each milestone has been met at the beginning of each week. Project status reports will be due weekly on Sundays at 1200 after the submission of your team's proposal.

You must have at least one major milestone identified per developer each week. Any successful software development effort always has a well-developed plan with organized tasks and milestones.

3.11 Related Projects

Research and identify any projects with similar goals that already exist in the open market or literature. You should be able to explain the most important aspects of those projects and how they differ from the proposed effort. Overlapping with one or more related projects is allowed for this assignment, but your project should provide some unique aspect or direction.

4 Programming Task

For this assignment, your team must show evidence that the proposed software development task is feasible. This is best achieved by setting up the software development framework and showing initial results with the tools and libraries required to complete the project.

Team Assignment 02 is an initial stage of development for your final project. Your submission should represent a substantial amount of progress toward achieving the final project goals. A simple demonstration showing “hello world” or the numerical equivalent is not sufficient. As part of your proposal, you should include preliminary results that include the start of your user interface, the solution to a simplified version of your problem, and/or a graphical display of your data.

There are certain common traits that all final projects must have:

- **Threads and concurrency.** Your program must incorporate multi-threading and be “thread-safe.” Thread-safe means that your program manipulates shared data structures without conflicts between threads trying to simultaneously access the data. Use of threads needs to be designed into the software architecture rather than added as an afterthought. You should be able to demonstrate a quantifiable speed-up in your algorithm computation when compared with using only a single core. Running jobs with a specified number of cores/GPUs on the CCV should allow you to test this acceleration and generate a table for the final project report.
- **Error handling.** Code should detect errors and handle them appropriately. If the program contains enough context to resolve a problem without user intervention, it should correct it, warn the user, and then move on. If there is invalid input, provide the user with an opportunity to correct for this type of error and resume computation. When errors cannot be corrected or resolved by user intervention, a meaningful error message should be provided to elucidate the problem to a human. Exceptions should be used over assertion errors. When necessary, design your own exception classes and use try..catch statements to detect such problems.
- **Graphical interface and/or data visualization.** Every final project must have a graphical component to it. The program should either 1) make use of a graphical user interface (GUI) designed with Qt to control the program in a meaningful way, and/or 2) visualize complex data sets using VRML, VTK, or another data visualization library. You may want to consider including both a GUI and data visualization to your project unless there is some suitably difficult task that each team member is undertaking for the final project.
- **Differential equations.** Since you are focusing on a scientific computing problem, it is expected that your final project includes numerical solutions to sets of differential equations. Using existing ODE/PDE solvers is acceptable and encouraged, but the complexity of the project should be non-trivial when doing so. If you have an alternative idea for a scientific computing problem (e.g., linear systems, machine learning, etc.) that doesn’t involve solving sets of differential equations, this may be allowed with permission.
- **Level of Difficulty.** The final project is expected to be non-trivial. You have a team of three developers, so logically dividing a complex problem into smaller achievable parts is essential. No one team member should be doing the bulk of the workload. Likewise, no one team member should take on only the simpler tasks. There should be a fair and equitable workload distributed across the project that all team members agree with. If one task proves to be much more difficult than another, team members should redirect their attention to assist or re-scope the project, as needed.

5 Grading

This assignment is worth 30% of the final project grade. Grading for this assignment will have a single grade for the team submission, but each individual is required to submit a self/team evaluation via CATME to receive any credit. Individual grades may be weighted (positive or negative) by the CATME team evaluation in cases where there is a clear bias in workload on one or two individuals.

5.1 Written Proposal (50%)

Your written proposal will be graded by the instructor and TA based on a rubric for technical writing. This rubric is available on canvas under the team assignment 02 page.

5.2 Programming Task (50%)

Your software development efforts will be graded on answers to the following questions:

- Does your team present sufficient evidence that the proposed software development task is feasible within the time constraints?
- Does your code have a software development framework/environment that will scale with the project?
- Has your team done sufficient preliminary experimentation with the tools and libraries required to complete the project?
- Does your code include a comprehensive set of built-in tests to validate the software is functioning as expected?
- Does your team make use of good software code management principles (i.e., frequent/daily use of a git repository, branches, merges, etc.)?
- Does your software build and run successfully from a git clone using CMake with the specified modules (provide a CMakeLists.txt file and a .modules file in your repository)?
- Does your repository include comprehensive build and usage instructions located in a README.txt file (or similar format)?

The software development effort for this assignment will also be graded on some basic software quality metrics:

- Consistent, easy to follow code structure (even across different developers – decide as a group on a standard format)
- Logically organized code partitioned into multiple functions, classes, and files
- Proper use of include header guards
- Ubiquitous descriptive comments in the code

Future software deliverables (weekly progress reports and source code repositories) will be graded on the above criteria, but also on meeting the expected goals set forth in the written proposal.

5.3 Peer Review

Your team's written proposal will be shared with the class on Canvas after being submitted for credit. This will allow other students to benefit and learn from your work.

5.4 Team Evaluations

This team assignment will have a follow-on survey using CATME to provide both positive feedback and constructive criticism to each member. The survey will be a self-assessment as well as an objective assessment of each of your team members. Please keep in mind that this feedback will be provided to the individuals you are assessing, and any negative feedback should remain constructive in nature. In this manner, critical feedback will help you and your team members identify some key areas to improve upon for the final project.