**Project**

Due: Thursday 14th October 2021 at 6pm

The aim of the course project is to give you an opportunity to synthesize topics you have learned during the Object-oriented Programming Course into a single system embedded in a real-world engineering challenge. You should have already selected a topic for your project. There are three deliverables: (1) functioning program, (2) report, and (3) self-assessment. You should study the provided evaluation scheme to understand how your work will be assessed.

1. **Program** (15 points): your program should be implemented to solve in a simplistic way your chosen project topic. It should not need to be overly complicated, but it should demonstrate optimizations for speed and memory, as well as object-oriented programming paradigms. The code should be readable, with a clear output to the console.
2. **Report** (18 points): your report should provide an educated reader with enough information to understand the challenge your program attempts to solve, how your program is structured, and the reason behind the decisions you made.
   * You should structure your report with headings/sub-headings that include an “introduction”, your “approach”, and your “solution”.
   * You should also include a diagram (high-level flowchart or UML diagram) that defines how your program is structured (e.g. using <draw.io>).
   * You should also highlight implementation of optimizations and object-oriented paradigms through the use of code snippets (do no copy and paste large sections of code).
   * A length of 3 to 6 pages in single-spaced 12pt font is reasonable for a report. *Please do not plagiarize*.
3. **Self-assessment** (2 points): you should study and submit the evaluation scheme below along with your report. It defines how you will be evaluated and what constitutes excellent work.

*Make sure to upload all three components with your netID as part of the filenames.*

**Evaluation Scheme**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 3 points | 2 points | 1 point | 0 point | Student self-evaluation | Instructor’s Evaluation |
| Code (15 points) | | | | | | |
| *Readability* | The code is exceptionally well organized and very easy to follow, with extensive use of comments, functions, and files to organize code. | The code is fairly easy to read, with moderate use of comments, functions, and files to organize code. | The code is readable only by someone who knows what it is supposed to be doing. | The code is poorly organized and very difficult to read. |  |  |
| *Optimization*  *Indicate file(s) and line number(s).* | Code implements a number of coding paradigms / topics covered in the class that aid in code optimizations for speed and memory. | Code implements a few coding paradigms / topics covered in the class that aid in code optimizations for speed and memory. | Code implements one coding paradigms / topics covered in the class that aid in code optimizations for speed and memory. | Code implements no coding paradigms / topics covered in the class that aid in code optimizations for speed and memory. |  |  |
|  | | | |
| *Object-oriented Paradigms*  *Indicate file(s) and line number(s).* | Code implements a number of object-oriented coding paradigms / topics covered in the class. | Code implements a few of object-oriented coding paradigms / topics covered in the class. | Code implements a few of object-oriented coding paradigms / topics covered in the class. | Code implements no object-oriented coding paradigms / topics covered in the class. |  |  |
|  | | | |
| *Output* | Code functionality prints very clearly to the console with good use of headers and newlines. | Code functionality prints to the console with fair use of headers and newlines. | Code functionality rarely prints to the console. | Code does not print to console to demonstrate functionality. |  |  |
| *Correctness* | Code runs perfectly with no errors. | Code runs with a few errors. | Code runs with several errors. | Code does not run at all. |  |  |
| Report (18 points) | | | | | | |
| *Engaging* | Writing was deeply engaging, with extensive use of real-word examples, suggesting deep thought in content delivery. | Writing was moderately engaging, with some reference to real-world examples. | Writing made no use of real-world examples. | Writing was difficult to follow and composed in a rush. |  |  |
| *Grammar* | Grammar is perfect. | Some number of grammatical errors. | Significant number of grammatical errors, indicating work was reviewed but not carefully. | Full of grammatical errors, indicating work was only a first draft. |  |  |
| *Structure* | Extensive use of spacing between paragraphs, headings, sub-headings. | Moderate use of spacing between paragraphs, headings, sub-headings. | Limited use of spacing between paragraphs, headings, sub-headings. | No use of spacing between paragraphs, headings, sub-headings. |  |  |
| *Visual* | Excellent use of flowchart / diagram to present logic of code system. | Fair use of flowchart / diagram to present logic of code system. | Poor use of flowchart / diagram to present logic of code system. No clear structure, use of color scheme, with unreadable text. | No use of flowchart / diagram to present logic of code system. |  |  |
| *Content* | Report highlights *all* key elements of code, and presents code snippets within document to help describe system. | Report highlights *most* key elements of code, and presents code snippets within document to help describe system. | Report highlights *some* key elements of code, and *poorly* presents code snippets within document in an attempt to describe system. | Report does not explicitly highlight key elements of code, with either no code snippets, or inserts a dump of the whole code base with little effort. |  |  |
| *References* | Significant number of references cited in bibliography and writing which may include articles, code blogs/repos, news stories, and website. | A number of references cited. | Almost no references. | No references cited, and writing makes claims that should be supported by references. |  |  |