**Faculty of Engineering, Environment and Computing**

##### 5003CEM Advanced Algorithms

**Assignment Brief 2022/23**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module Title  **Advanced Algorithms** | Ind/Group  **Individual** | | Cohort (Sept/Jan/May)  **Sept** | Module Code  **5003CEM** |
| Coursework Title  **CW: Code Submission and Report** | | | | Hand out date:  **16.01.2023** |
| Lecturer  **John Halloran / Beate Grawemeyer** | | | | Due date:  **06.04.2023 18:00** |
| Estimated Time (hrs): **50**  Word Limit\*: 3000 | | Coursework type:  **Code Submission / Report** | | % of Module Mark  **66.66% (10 of 15 credits)** |
| Submission arrangement online via Aula: **Upload is a (zipped) folder of code and a report.**  File types and method of recording: **Python and / or C++ files, plus report**  Mark and Feedback date: **within 3 working weeks of submission.**  Mark and Feedback method: **Rubric marks and individualized comments.** | | | | |

|  |
| --- |
| **Module Learning Outcomes Assessed:**  2 Design and implement algorithms and data structures for novel problems  3 Specify and implement methods to estimate solutions to intractable problems   1. Design and implement a basic concurrent application |
| **SUBMISSION**  You will submit your Assignment as a document with a separate codebase, as described above.  **The deadline is Thursday 6th April 2023: 18:00** |
| **ASSIGNMENT BRIEF**  CODE SUBMISSION  Each week on 5003CEM you will be given a range of programming tasks.  Some of these are defined as assessed, some as non-assessed.  Assessed tasks are required work, which you must submit by April 6th 2023 at 18:00.  There will be EIGHT assessed tasks to be completed. 5 of these will be at a standard level of difficulty; and 3 will be at an advanced level of difficulty. Each of the advanced tasks will relate to each of the 3 learning outcomes specified.  The tasks will cover a range of algorithms taught on the module, as well a basic concurrent application. You will also need to consider novel algorithms.  Assessed tasks will be evenly spaced across the module from Weeks 4 to 9.  Your assessed tasks will all need to be kept within the same folder with (a) nothing else in it; and (b) a clear structure with appropriately named files (and folder).  There will be a Aula submission, in two parts: one for a zip of your code, and the second for your report.  MARKING CRITERIA  Over. |
| Notes   1. If relevant, you are expected to use the [CUHarvard](https://curve.coventry.ac.uk/open/file/bdfb947c-9d43-48d3-8ec8-f511682e1dd1/1/The%20CU%20Guide%20to%20Referencing%20in%20Harvard%20Style.pdf) referencing format. For support and advice on how this students can contact [Centre for Academic Writing (CAW)](http://www.coventry.ac.uk/study-at-coventry/student-support/academic-support/centre-for-academic-writing/?theme=main). 2. Please notify your registry course support team and module leader for disability support. 3. Any student requiring an extension or deferral should follow the university process as outlined [here](https://share.coventry.ac.uk/students/Registry/Pages/Deferrals-and-Extension.aspx). 4. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computer. Students should therefore regularly back-up any work and are advised to save it on the University system. 5. If there are technical or performance issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours or the next working day if the deadline falls on a Friday or over the weekend period. This will be communicated via email and as a CUMoodle announcement. |

**Matrix marking scheme (marking criteria by degree band)**

|  |  |  |
| --- | --- | --- |
|  | **CODE SUBMISSION**  **50 MARKS** | **REPORT**  **50 MARKS** |
| Marking criterion | **Code quality**  **/50** | **Knowledge of code**  **/50** |
| 1st (70+) | Code is fully working, bug-free, well-commented. | Student shows sophisticated and detailed knowledge of how the code works at every level. Diagrams and worked examples are likely to feature. Code is pasted into the report properly formatted. Student can explain the design and implementation decisions. May offer alternative implementations. There is strong critical comment using clear criteria which may include some or all of the following: complexity, conciseness, readability, language choice, implementation issues and choices. |
| 2.1 (60-69) | Code works, may have one or two non-important bugs. Commented. | Good knowledge of how major sections of the code work. Diagrams and worked examples will probably feature. Code is pasted into the report properly formatted. May offer some alternative implementations. Is able to explain design decisions and why things were implemented in particular ways. There will be some critical comment as above. |
| 2.2 (50-59) | Code generally works, may have bugs. Has some comments. | Some knowledge of how the code works but there may be areas of confusion. May be some diagrams / worked examples but incomplete and possibly showing misconceptions. Code is pasted into the report but may not be properly formatted. Less likely to offer alternative implementations. Ability to explain basic decisions about the design and implementation but again there may be misconceptions. Unlikely to be much if any critical comment. |
| 3rd (40-49) | Code may not work; may have significant bugs; but shows potential to work. May have some comments. | Limited knowledge of how the code works which is unlikely to show insight at detailed levels. Unlikely to be many diagrams or worked examples and these may be less effective. Unlikely to offer alternative implementations. Some code may be pasted into the report but may not be properly formatted. Aware of the design of the code but not much insight into different design decisions that could be made. Little or no critical comment. |
| Fail (0-39) | At this level some or all of the following may apply: Code is incomplete, does not work, is not commented, does not show clear potential. At the zero end, nothing has been done. | Little if any ability to explain how the code works. Unlikely to be worked examples or diagrams. No alternative implementations. May be some code pastes but poorly formatted. No critical comment. At the zero level the explanation (if any) will have no value. |

PAGE LEFT INTENTIONALLY BLANK