# UCL School of ManagementCoursework Assessment Brief

|  |  |
| --- | --- |
| **Module code/name** | MSIN0023 Computational Thinking |
| **Module leader name** | Matt Jones (Cohorts A/C), Andrew Whiter (Cohort B) |
| **Academic year** | 2021/22 |
| **Term** | 1 |
| **Assessment title** | Assignment 2: Investigate the use of an algorithm to solve a specific business problem |
| **Individual/group assessment** | Individual |

**Return and status of marked assessments:** Within 4 weeks from the date of submission as per UCL guidelines. The module team will update you if there are delays through unforeseen circumstances (e.g. ill health). All results when first published are provisional until confirmed by the Examination Board.

**Copyright Note to students:** Copyright of this assessment brief is with the module leader(s) named above. If this brief draws upon work by third parties (e.g. Case Study publishers) such third parties also hold copyright. It must not be copied, reproduced, transferred, distributed, leased, licensed or shared any other individual(s) and/or organisations, including web-based organisations, without permission of the copyright holder(s).

**Academic Misconduct:** Academic Misconduct is defined as any action or attempted action that may result in a student obtaining an unfair academic advantage. **Academic misconduct includes plagiarism, obtaining help from/sharing work with others be they individuals and/or organisations or any other form of cheating.** Refer to [Academic Manual Section 9: Student Academic Misconduct Procedure - 9.2 Definitions](https://www.ucl.ac.uk/academic-manual/chapters/chapter-6-student-casework-framework/section-9-student-academic-misconduct-procedure#9.2).

**Referencing:** You must reference and provide full citation for ALL sources used, including articles, text books, lecture slides and module materials. This includes any direct quotes and paraphrased text. If in doubt, reference it. If you need further guidance on referencing please see UCL’s referencing tutorial for students here: <https://library-guides.ucl.ac.uk/referencing-plagiarism/welcome.> Failure to cite references correctly may result in your work being referred to the Academic Misconduct Panel.

# Content of this assessment brief

|  |  |
| --- | --- |
| Section | Content |
| A | **Core information** |
| B | **Coursework brief and requirements** |
| C | **Module learning outcomes covered in this assessment** |
| D | **Groupwork instructions (if applicable)** |
| E | **How your work is assessed** |
| F | **Additional information** |

# 

# Section A: Core information

|  |  |
| --- | --- |
| **Submission date** | 24/11/2021 |
| **Submission time** | 4pm |
| **Assessment is marked out of:** | 100 |
| **% weighting of this assessment within total module mark** | 20% |
| **Maximum word count/page length/duration** | The maximum word count for this assignment is 2000 words. |
| **Footnotes, appendices, tables, figures, diagrams, charts included in/excluded from word count/page length?** | Footnotes are included in the word count, as well as any substantive text in any tables, figures, diagrams, or charts.  The list of code sources in the appendix is excluded from word count. |
| **Bibliographies, reference lists included in/excluded from word count/page length?** | Bibliographies are excluded from the word count. |
| **Penalty for exceeding word count/page length** | Standard UCL penalties for exceeding (deduction of 10 percentage points, capped at 40% for Levels 4,5, 6, and 50% for Level 7) Refer to Academic Manual Section 3: Module Assessment - 3.13 Word Counts. |
| **Penalty for late submission** | Standard UCL penalties apply. Students should refer to Refer to <https://www.ucl.ac.uk/academic-manual/chapters/chapter-4-assessment-framework-taught-programmes/section-3-module-assessment#3.12> |
| **Submitting your assessment** | The assignment MUST be submitted to the module submission link located within this module’s Moodle ‘Submissions’ tab by the specified deadline. |
| **Anonymity of identity. Normally, all submissions are anonymous unless the nature of the submission is such that anonymity is not appropriate, illustratively as in presentations or where minutes of group meetings are required as part of a group work submission** | The nature of this assessment is such that anonymity is required. |

# 

# Section B: Assessment Brief and Requirements

|  |
| --- |
| Investigate the use of an algorithm to solve a specific business problem This assignment has been designed to help prepare you for your final assignment due in February 2022 (Assignment 3), which also requires you to start by identifying a business problem that can be addressed using algorithmic software. (However, for Assignment 3, you will also be required to develop or reuse existing python code, analyse and comment on this code, and test it on data you have generated or found for your chosen business problem.)  While you can, later, choose a different business problem and algorithm for Assignment 3, you are encouraged to use the same problem and algorithm for both Assignment 2 and Assignment 3. This will provide you with good starting foundation for Assignment 3, as well as the opportunity to receive some feedback on your choice of problem and algorithm.  **Note that for this assignment, you do not need to understand the exact details of how your chosen algorithm works or is coded.**  Instructions:  Identify and investigate a business problem whose solution involves a **non-trivial algorithm**. Non-trivial algorithms must include iterations (i.e. loops or recursion) and be able to process data inputs of arbitrary size (eg datasets/lists/matrices of any size).  While you will need to define a hypothetical scenario for your business problem (see below) you do not need to identify or research a specific company. Your problem should be a general class of problem typically faced by any business in a given sector.  Examples of business problems addressed using non-trivial algorithms include:   * Operations optimisation: production planning, routing planning, resource optimisation, workforce scheduling, maintenance scheduling, call centre queue management … * Marketing/Sales: market data analysis, predicting customer buying behaviour * Finance/Trading: forecasting, financial analysis, trade optimisation * Online Business: recommendations, search * Industry/Product Specific: facial recognition, traffic management, and many more …   Whilst you can consider any algorithm, choosing an algorithm you have already covered (or will cover in Computational Thinking) is likely to help focus your thinking, eg   * Gauss-Jordan Elimination from Mathematics II * Classification Tree Partitioning from Data Analytics I * Knapsack Packing or Breadth-First Search from Computational Thinking  (you can find out much more about these and other algorithms in your textbook: *Introduction to Computation and Programming Using Python*, John V. Guttage). * Simplex Linear Programming from Computational Thinking   Your report should have the following 4 sections and answer the following questions:   1. **Nature and history of your algorithm**    * Briefly, describe – in your own words - your algorithm and how it works. This does not need to be a detailed description of how the algorithm works. (Do not simply copy & paste definitions. You must explain any technical terms you use.)    * Investigate and outline the history of your algorithm? Who invented the algorithm? When? What problem was it originally used to solve? How has it since been refined?    * Besides the problem you are focusing on, what other applications/problems has your algorithm be used to solve?    * Investigate and discuss any cited pros/cons of using this algorithm? How do these pros/cons relate to the nature and scale of the problem being solved? 2. **Your business problem**    * Briefly, describe your chosen problem. What industry are your focusing on? Why is your problem particularly associated with this industry?    * What business decision is being made?    * How does solving your problem impact the business?    * Describe how this business problem can be decomposed (broken down) into smaller sub-problems?    * Consider the ways in which the problem can be generalised. What are the simplest or most common versions of this business problem? What might make this problem more complex to solve? What unusual/extreme versions of this problem can you envisage or find? 3. **Applying your algorithm to your business problem**     * Briefly, describe a hypothetical scenario for your business problem.    * What aspect of the business problem does your algorithm particularly address in this scenario?    * Describe the nature of the data being used by the algorithm in this scenario? How much data is there? How often is this data added to or updated?    * Describe how often the algorithm needs to run in this scenario, eg continuously, daily, monthly, yearly?    * Discuss how quickly your algorithm needs to provide a solution in your scenario? Also, discuss how necessary it is for the solution to be optimised. If not, how might you characterise a “good enough” solution?    * Investigate, describe and reference any real-life examples of your chosen algorithm being used to solve your chosen business problem. If you are unable to find such examples, discuss why this may be.    * Are there other algorithms that could be used to alternatively solve your business problem? What advantages/disadvantages do these alternatives have compared to your algorithm? 4. **Code source appendix**   Include in an appendix, a list of references to potential sources of python code that you have been able to identify that implement your algorithm. (Do not include any code.) Try to identify examples of python code that implement your algorithm “from scratch”, i.e. without using any non-core python modules. Do not worry at this stage about trying to understand any code. |

# Section C: Module Learning Outcomes covered in this Assessment

|  |
| --- |
| This assessment contributes towards the achievement of the following stated module Learning Outcomes as highlighted below:   * Understand how modern algorithmic software technologies add value to businesses. * Understand how to express and design algorithms in terms of sequences of instructions, selection rules and iterative processes. * Characterise algorithms and the data structures they employ. |

# Section D: Groupwork Instructions (where relevant/appropriate)

|  |
| --- |
| Not applicable. |

# Section E: How your work is assessed

|  |
| --- |
| Within each section of this assessment you may be assessed on the following aspects, as applicable and appropriate to this assessment, and should thus consider these aspects when fulfilling the requirements of each section:   * The accuracy of any calculations required. * The strengths and quality of your overall analysis and evaluation; * Appropriate use of relevant theoretical models, concepts and frameworks; * The rationale and evidence that you provide in support of your arguments; * The credibility and viability of the evidenced conclusions/recommendations/plans of action you put forward; * Structure and coherence of your considerations and reports; * Appropriate and relevant use of, as and where relevant and appropriate, real world examples, academic materials and referenced sources. Any references should use either the Harvard OR Vancouver referencing system (see [References, Citations and Avoiding Plagiarism](https://www.ucl.ac.uk/library/docs/guides/references-plagiarism)) * Academic judgement regarding the blend of scope, thrust and communication of ideas, contentions, evidence, knowledge, arguments, conclusions. * Each assessment requirement(s) has allocated marks/weightings.   Student submissions are reviewed/scrutinised by and internal assessor and are available to an External Examiner for further review/scrutiny before consideration by the relevant Examination Board.  It is not uncommon for some students to feel that their submissions deserve higher marks (irrespective of whether they actually deserve higher marks). To help you assess the relative strengths and weaknesses of your submission please refer to UCL Assessment Criteria Guidelines, located at <https://www.ucl.ac.uk/teaching-learning/sites/teaching-learning/files/migrated-files/UCL_Assessment_Criteria_Guide.pdf>  The above is an important link as it specifies the criteria for attaining 85% +, 70% to 84%, 60% to 69%, 50% to 59%, 40% to 49%, below 40%.  You are strongly advised to **not** compare your mark with marks of other submissions from your student colleagues. Each submission has its own range of characteristics which differ from others in terms of breadth, scope, depth, insights, and subtleties and nuances. On the surface one submission may appear to be similar to another but invariably, digging beneath the surface reveals a range of differing characteristics. |

# Section F: Additional information from module leader (as appropriate)

|  |
| --- |
| None. |