Logo: Cardiff Metropolitain University

School of xx

### Assessment

### Brief

|  |  |
| --- | --- |
| Module Code | Module Title |
| CIS6007 | Parallel and Distributed Systems |
| Academic Year | Semester |
| 21/22 | 2 |
| Module Leader email | |
| [tmallikarachchi@cardiffmet.ac.uk](mailto:tmallikarachchi@cardiffmet.ac.uk) | |

Content

[Assessment Details 2](#_Toc67665314)

[Submission Details 3](#_Toc67665315)

[Assessment Criteria 3](#_Toc67665316)

[Further Information 5](#_Toc67665317)

[Who can answer questions about my assessment? 5](#_Toc67665318)

[Referencing 5](#_Toc67665319)

[Submission problems 5](#_Toc67665320)

[Unfair academic practice 5](#_Toc67665321)

[How is my work graded? 6](#_Toc67665322)

# Assessment Details

|  |  |  |
| --- | --- | --- |
| Assessment title | Abr. | Weighting |
| Multithreaded programming assignment | WRIT1 | 50% |
| Pass marks are 40% for undergraduate work and 50% for postgraduate work unless stated otherwise. | | |

|  |  |
| --- | --- |
| Task/assessment brief: | |
| **Part 1) 70%**  You are required to implement a multithreaded image classification application that uses a K-Nearest Neighbour (KNN) algorithm. Following tasks should be completed during the process.   1. Identify a suitable dataset. You may use resources such as Kaggle, MNIST, or CIFAR websites for this purpose. Helpful links will be provided in Moodle. You should discuss the suitability of the dataset with the teaching team before using it for this task. Dataset should have sufficient number of images to split it into training and testing images. 2. Investigate the methods that can be used to read the and visualize the data using C++. You may have to use 3rd party libraries such as OpenCV or follow customized binary data reading mechanisms provided in the documentation of the dataset. 3. Use appropriate data structures, and variables to store the image, and its label within the program. 4. Once images are loaded implement functionality in the console application to provide a new image from the testing dataset. You may extend the options to provide correct categorization of the image to test your kNN algorithm. There are two options:    1. Provide a single image as a testing image    2. Provide the path to a folder containing a collection of images 5. Once the training images (and labels) and testing image (s) are loaded, the program should execute kNN classification to classify the test image(s). If you have provided a single image, the program should output the image name, its classification, confidence level of the result, and execution time. If you have provided a test image folder, the program should output the classification results for all images in the folder, their confidence levels and execution time. 6. You should generate a report along with your implementation describing the following:    1. Details of the dataset. Image resolutions, format, and additional meta data    2. Data structures used to store the data    3. Synchronization methods used throughout the program and the reasons for selecting them    4. Parallel programming patterns/strategies used when designing the multithreaded application    5. Execution performance of the application should be compared against the serial version of the program. You should compute speed-up achieved in following cases:       1. Increasing number of threads used in the program       2. Use of different data structures e.g., vectors, lists, or custom arrays to store the training/testing images in the program   **Pseudo code for the kNN algorithm for testing a single image**  train\_dataset := [train\_images, train\_lables] *# structure that holds training images and labels*  d\_img; *# Variable that holds test image*  distance\_lbl\_array; *# A structure to hold distances and labels*    *# iterate through each element in the train\_dataset*  for each element *e* in train\_dataset  t\_img := train\_dataset[e][0] *# eth training image*  t\_lbl := train\_dataset[e][1] *# eth label for the corresponding image*    *# compute the Euclidian distance between training image and test image*  # *store the* *Euclidian distance and the label of the training image*  distance\_lbl\_array[e] := [distance, t\_lbl]  end    *# Sort distance\_lbl\_array according to the distance*  sorted\_distance\_lbl\_array := sort (distance\_lbl\_array)    *# extract elements for k closest labels. e.g., if k=10, get the first 10 elements in the sorted\_distance\_lbl\_array.*  closest\_k\_label\_array := sorted\_distance\_lbl\_array[:k]    *# Find the label class with maximum number of label counts*  Estimated\_label := max\_label\_count(closest\_k\_label\_array)    *# Find the confidence as a ratio of selected label count to total number of closest elements*  Confidence := max\_label\_count(closest\_k\_label\_array)/size(closest\_k\_label\_array)  **Part 2): 20%**  Lab demonstrations on individual worksheets. You will be required to carry out code demonstrations on weekly exercises in the workshop activities.  **Part 3): 10%**  A screen recorded video demonstrating the full functionality of the part 1) task should be uploaded to the dedicated Moodle submission point. The recording should demonstrate the compilation steps, execution, and outputs of the program. | |
| Word count (or equivalent): | 2000 |
| This a reflection of the effort required for the assessment. Reference lists and contents of appendices are excluded from the word count. Contents of appendices are not usually considered when determining your final assessment grade.  The report should focus on part 1) assessment task described in the assessment brief. You may use graphs, figures, and plots where necessary. | |

|  |
| --- |
| Academic or technical terms explained: |
| N/A |

# Submission Details

|  |  |  |  |
| --- | --- | --- | --- |
| Submission Deadline: | This will be provided on the Moodle submission point. | Estimated Feedback  Return Date | This will normally be 20 working days after initial submission. |
| Submission  Time: | By 4.00pm on the deadline day. |  | |
| Moodle/Turnitin: | **Any assessments submitted after the deadline will not be marked and will be recorded as a non-attempt unless you have had an extension request agreed or have approved mitigating circumstances. See the School Moodle pages for more information on extensions and mitigating circumstances.** | | |
| File Format: | The source code and report should be submitted as a .zip file. Source files available in git will NOT be considered for marking.  A video clip demonstrating the full functionality of the part 1) and part 2) tasks should uploaded as a media file. If the file size exceeds the Moodle limit, you will be required to upload the video to onedrive and provide the link with correct permissions.  **Your assessment should be titled with your:**  **student ID number, module code and assessment ID,**  **e.g. st12345678 BHL5007 WRIT1** | | |
| Feedback | Feedback for the assessment will be provided electronically via Moodle. Feedback will be provided with comments on your strengths and the areas which you can improve. View the [guidance](https://learn.cardiffmet.ac.uk/mod/glossary/showentry.php?courseid=8107&eid=9581&displayformat=dictionary) on how to access your feedback.  All marks are provisional and are subject to [quality assurance processes](https://outlookuwicac.sharepoint.com/:b:/s/QED/Ec3kYQQeEHdKrCbo_tJnr2kBomIiiLINmPebUgvTUljq9Q?e=a0G2z5) and confirmation at the programme Examination Board. | | |

# Assessment Criteria

|  |
| --- |
| Learning outcomes assessed |
| * Implement appropriate parallel and distributed software solutions to common problems found in computationally heavy and data-driven domains * Critically evaluate the fundamental issues in the design of distributed algorithms, protocols and systems, such as timing, coordination and consensus |
| Other skills/attributes developed  This includes elements of the Cardiff Met EDGE (Ethical, Digital, Global and Entrepreneurial skills) and other attributes developed in students through the completion of the module and assessment. These will also be highlighted in the module guidance, which should be read by all students completing the module. Assessments are not just a way of auditing student knowledge. They are a process which provides additional learning and development through the preparation for and completion of the assessment. |
| * Advanced computing skills, * Problem solving and programming skills |

|  |
| --- |
| Marking/Assessment Criteria |
| |  |  | | --- | --- | | **Part 1)** | **70%** | | Implementation and use of proper programming techniques | 20 | | use of best practices, version controlling | 10 | | Use of threads | 10 | | Use of appropriate data structures | 10 | | Use of proper synchronization techniques | 10 | | Report | 10 | |  |  | | **Part 2) Lab demonstrations** | **20%** | | **Part 3) Video demonstration of part 1)** | **10%** | | **Total** | **100%** | |

# Further Information

*Staff note – Please check that these are aligned with your programme’s expectations.*

## Who can answer questions about my assessment?

Questions about the assessment should be directed to the staff member who has set the task/assessment brief. This will usually be the Module Leader. They will be happy to answer any queries you have.

Staff members can often provide feedback on an assignment plan but cannot review any drafts of your work prior to submission. The only exception to this rule is for Dissertation Supervisors to provide feedback on a draft of your dissertation.

## Referencing and independent learning

Please ensure you reference a range of credible sources, with due attention to the academic literature in the area. The time spent on research and reading from good quality sources will be reflected in the quality of your submitted work.

Remember that what you get out of university depends on what you put in. Your teaching sessions typically represent between 10% and 30% of the time you are expected to study for your degree. A 20-credit module represents 200 hours of study time. The rest of your time should be taken up by self-directed study.

Unless stated otherwise you must use the **HARVARD** referencing system. Further guidance on referencing can be found in the Study Smart area on Moodle and at [www.citethemrightonline.com](http://www.citethemrightonline.com) (use your university login details to access the site). Correct referencing is an easy way to improve your marks and essential in achieving higher grades on most assessments.

## Technical submission problems

It is strongly advised that you submit your work at least 24 hours before the deadline to allow time to resolve any last minute problems you might have. If you are having issues with IT or Turnitin you should contact the IT Helpdesk on (+44) 2920 417000. You may require evidence of the Helpdesk call if you are trying to demonstrate that a fault with Moodle or Turnitin was the cause of a late submission.

## Extensions and mitigating circumstances

Short extensions on assessment deadlines can be requested in specific circumstances. If you are encountering particular hardship which has been affecting your studies, then you may be able to apply for mitigating circumstances. This can give the teachers on your programme more scope to adapt the assessment requirements to support your needs. Extensions and mitigating circumstances policies and procedures are regularly updated. You should refer to your degree programme or school Moodle pages for information on extensions and mitigating circumstances.

## Unfair academic practice

Cardiff Met takes issues of unfair practice **extremely seriously.** The University has procedures and penalties for dealing with unfair academic practice. These are explained in full in the University's Unfair Practice regulations and procedures under [Volume 1, Section 8](https://www.cardiffmet.ac.uk/registry/academichandbook/Pages/Ah1_08.aspx) of the Academic Handbook. The Module Leader reserves the right to interview students regarding any aspect of their work submitted for assessment.

Types of Unfair Practice, include:

**Plagiarism,** which can be defined as using without acknowledgement another person’s words or ideas and submitting them for assessment as though it were one’s own work, for instance by copying, translating from one language to another or unacknowledged paraphrasing. Further examples include:

* Use of any quotation(s) from the published or unpublished work of other persons, whether published in textbooks, articles, the Web, or in any other format, where quotations have not been clearly identified as such by being placed in quotation marks and acknowledged.
* Use of another person’s words or ideas that have been slightly changed or paraphrased to make it look different from the original.
* Summarising another person’s ideas, judgments, diagrams, figures, or computer programmes without reference to that person in the text and the source in a bibliography/reference list.
* Use of assessment writing services, essay banks and/or any other similar agencies (NB. Students are commonly being blackmailed after using essay mills).
* Use of unacknowledged material downloaded from the Internet.
* Re-use of one’s own material except as authorised by your degree programme.

**Collusion**, which can be defined as when work that that has been undertaken with others is submitted and passed off as solely the work of one person. Modules will clearly identify where joint preparation and joint submission are permitted, in all other cases they are not.

**Fabrication of data**, making false claims to have carried out experiments, observations, interviews or other forms of data collection and analysis, or acting dishonestly in any other way.

## How is my work graded?

Assessment grading is subject to thorough quality control processes. You can view a summary of these processes on the [Assessment Explained Infographic](https://outlookuwicac.sharepoint.com/sites/QED/Shared%20Documents/Forms/Front%20Page.aspx?id=%2Fsites%2FQED%2FShared%20Documents%2Fstudent%20guide%20%2D%20Is%20my%20mark%20fair%2Epdf&parent=%2Fsites%2FQED%2FShared%20Documents&p=true&originalPath=aHR0cHM6Ly9vdXRsb29rdXdpY2FjLnNoYXJlcG9pbnQuY29tLzpiOi9zL1FFRC9FYzNrWVFRZUVIZEtyQ2JvX3RKbnIya0JvbUlpaUxJTm1QZWJVZ3ZUVWxqcTlRP3J0aW1lPXFLb08zblB3MkVn).

Grading of work at each level of Cardiff Met degree courses is benchmarked against a set of general requirements set out in [Volume 1, Section 4.3](https://www.cardiffmet.ac.uk/registry/academichandbook/Documents/AH1_04_03.pdf) of our Academic Handbook. A simplified version of these Grade Band Descriptors (GBDs) with short videos explaining some of the academic terminology used can be accessed via the [Facilitation of Learning](https://outlookuwicac.sharepoint.com/sites/QED/SitePages/Facilitation-of-Learning.aspx) resource page.

We would strongly recommend looking at the [Study Smart](https://learn.cardiffmet.ac.uk/course/view.php?id=1416) area of Moodle to find out more about assessments and key academic skills which can have a significant impact on your grades. Always check your work thoroughly before submission.

