EST 1892 LSBU

Coursework 1 Specification

CW1_Specification_CSI-4-DSA_21-22

Read this coursework specification carefully, it tells you how you are going to be assessed, how to submit your coursework on-time and how (and when) you'll receive your marks and feedback.

Module Code	CSI-4-DSA
Module Title	Data Structures and Algorithms
Lecturer	Mike Child
% of Module Mark	50%
Distributed	
Submission Method	Submit online via this Module's Moodle site
Submission Deadline	Friday 01/04/22 at 16:00
Release of Feedback & Marks	Feedback and provisional marks will be available in the Gradebook on Moodle from [28/04/22]

Coursework Aim:

Write a report on the activities you carried out in group using the Raspberry Pi and ultra-sonic rangefinder, particularly documenting the software that was developed (see *Assignment Task*, below).

Coursework Details:

Type:	Report
Word Count:	No formal word count is required but for general guidance no more than 2000 words would be expected. The word count will always be ambiguous because the requirement for embedded code will skew it. There are no specific penalties for exceeding or failing to reach the word count guidance, but your submission must properly address the task in hand.
Presentation:	▪ Work must be submitted as a word processor document (odt/doc/docx) or a PDF ▪ Your student number must appear at the front of the coursework. Your name must **not** be on your coursework.
Referencing:	While there is no specific requirement for references in this assignment, Harvard Referencing should be used for any you supply, see your Library Subject Guide for guides and tips on referencing.
Regulations:	Make sure you understand the University Regulations on expected academic practice and academic misconduct. Note in particular: ▪ Your work must be your own. Markers will be attentive to both the plausibility of the sources provided as well as the consistency and approach to writing of the work. Simply, if you do the research and reading, and then write it up on your own, giving the reference to sources, you will approach the work in the appropriate way and will cause not give markers reason to question the authenticity of the work. ▪ All quotations must be credited and properly referenced. Paraphrasing is still regarded as plagiarism if you fail to acknowledge the source for the ideas being expressed. **TURNITIN:** When you upload your work to the Moodle site it will be checked by anti-plagiarism software.

Learning Outcomes

This coursework will fully or partially assess the following learning outcomes for this module.

- Describe different algorithms and the means used to measure their performance.
- Analyse programming problems and identify appropriate algorithmic solutions.
- Develop software to solve relatively complex problems and assess alternative solutions.
- Apply critical and analytic reasoning to tasks.

Assignment Task

During this module you worked in a group to develop some software to implement a simple gesture detection system using a Raspberry Pi and an ultra-sonic range finder.

Your individual task is to write an explanation of how the program your group developed works in your own words.

- You must clearly explain the purpose of the code you have developed, including fragments of this code in the text of your report as you explain it. The report must be written as a proper explanation - **it is not sufficient to describe the code line-by-line**. You must explain the program as a series of problems being solved, with each problem broken down into smaller problems.
- You must identify the sections of code that address each problem and explain how it works to solve the problem. You must also try and explain why you used the approach you did and alternative ways of doing it you might have used.
- You must discuss the ways in which the program relates to the theoretical concepts covered in the module, in particular, which data structures have been used, how they work behind the scenes and the general performance characteristics you would expect them to have.
- You should document your use of reference materials both when working in the group and individually in your report to demonstrate your competence at finding information you need in technical documentation.

While the ***development*** of the code is intended to be a collaborative endeavour you work together with your group on, the report you submit must be written independently and demonstrate your own understanding of what the group has achieved. All reports will be subjected to similarity analysis to identify students who have not written in their own words but used the same writing as others.

Assessment Criteria and Weighting

LSBU marking criteria have been developed to help tutors give you clear and helpful feedback on your work. They will be applied to your work to help you understand what you have accomplished, how any mark given was arrived at, and how you can improve your work in future.

For this assignment the following criteria will be applied (also see rubric following).

Marking Criteria

This assignment will be marked using an adaptation of the University's standardised marking criteria. It is important that you pay attention to the criteria that will be applied and address them in the text of your report. A detailed rubric is shown on the next page, but the main criteria are as follows:

1. **Subject Knowledge (35%)**

   Understanding and application of subject knowledge. Contribution to subject debate.
   Assessed implicitly by your written explanation of the code, and explicitly by your ability to discuss it in terms of the theoretical content of the module and demonstrate understanding of data structures and the significance of the algorithms associated with them.
2. **Critical Analysis (15%)**

   Assessed by the rationale you give for your design approaches and their contrast to alternative approaches, and your evaluation of the finished program.
3. **Testing and Problem-Solving Skills (30%)**

   Design, implementation, testing and analysis of product / process / system / idea / solution(s) to practical or theoretical questions or problems.
   Assessed on the basis of the level of achievement of the software your group has developed together with its documentation, bearing in mind that code that you do not discuss in your narrative will be given very little credit (as the implication is that you do not understand what it is and what it does if you did not discuss it). However, your ability to explain the problems involved and the solutions they demanded, will also be considered here, whether or not the group was able to solve them. For this reason the marks awarded here are not simply based on what the group managed to achieve.
4. **Practical Competence (10%)**

   Skills to apply theory to practice or to test theory.
   Assessed on documented evidence of your use of technical documentation (for example Python tutorials and reference documentation and course materials) in working out how to accomplish the assignment.
5. **Personal and Professional Development (10%)**

   Management of learning through self-direction, planning and reflection
   Assessed on the basis of the quality of your submitted report, including clarity of writing, presentation, and properly addressing the assignment specification.

Please note the criteria weightings and general interpretation shown in bold capitals under each criteria.

Criteria	Outstanding 100-80%	Excellent 79-70%	Very good 69-60%	Good 59-50%	Satisfactory 49-40%	Inadequate 39-30%	Very poor 29-0%
Subject Knowledge Understanding and application of subject knowledge. Contribution to subject debate. **CODE EXPLANATION 35%**	Shows sustained breadth, accuracy and detail in understanding key aspects of subject. Contributes to subject debate. Awareness of ambiguities and limitations of knowledge.	Shows breadth, accuracy and detail in understanding key aspects of subject. Contributes to subject debate. Some awareness of ambiguities and limitations of knowledge.	Accurate and extensive understanding of key aspects of subject. Evidence of coherent knowledge.	Accurate understanding of key aspects of subject. Evidence of coherent knowledge.	Understanding of **key aspects** of subject. Some evidence of coherent knowledge.	Some evidence of superficial understanding of subject. Inaccuracies.	Little or no evidence of understanding of subject. Inaccuracies.
Critical Analysis Analysis and interpretation of sources, literature and/or results. Structuring of issues/debates. **RATIONALE FOR DESIGN APPROACHES 15%**	Outstanding demonstration of critical analysis of the possible design strategies that could be used to meet the software requirements, and evaluation of the approaches chosen.	Excellent demonstration of critical analysis of the possible design strategies that could be used to meet the software requirements, and evaluation of the approaches chosen.	Very good demonstration of critical analysis of the possible design strategies that could be used to meet the software requirements, and evaluation of the approaches chosen.	Good demonstration of critical analysis of the possible design strategies that could be used to meet the software requirements, and evaluation of the approaches chosen.	Demonstration of critical analysis of the **key** possible design strategies that could be used to meet the software requirements, and evaluation of the approaches chosen.	Trivial demonstration of critical analysis of the possible design strategies that could be used to meet the software requirements, and evaluation of the approaches chosen.	Little or no critical analysis has been demonstrated.
Testing and Problem-Solving Skills Design, implementation, testing and analysis of **product/process/system/idea/solution(s)** to practical or theoretical questions or problems **IMPLEMENTATION AND DISCUSSION 30%**	Outstanding implementation of all required software, with near perfectly organised, formatted and documented source code, and documented demonstration of runtime behaviour.	Excellent implementation of all required software, with well organised, formatted and documented source code provided	Competent implementation of all required software, with well organised, formatted and documented source code, and documented demonstration of runtime behaviour.	Implementation of all required software, with well organised, formatted and documented source code, and documented demonstration of runtime behaviour, with some missing/incorrect functionality or poor quality.	Implementation of most of the required software, with well organised, formatted and documented source code, and documented demonstration of runtime behaviour, with some missing/incorrect functionality or poor quality.	Implementation of only part of the required software, with well organised, formatted and documented source code, and documented demonstration of runtime behaviour, with some missing/incorrect functionality or poor quality.	Little or no functionality has been implemented.
Practical Competence Skills to apply theory to practice or to test theory **USE OF REFERENCE MATERIAL 10%**	Outstanding descriptions of factual information, programming techniques or theoretical explanations being found in technical or theoretical reference material.	Excellent explicit descriptions of all factual information, programming techniques or theoretical explanations that were found in technical or theoretical reference material.	Good explicit descriptions of all factual information, programming techniques or theoretical explanations that were found in technical or theoretical reference material.	Reasonable descriptions of most factual information, programming techniques or theoretical explanations that were found in technical or theoretical reference material.	Basic examples of the **main** factual information, programming techniques or theoretical explanations that were found in technical or theoretical reference material.	Some trivial examples of factual information, programming techniques or theoretical explanations being found in technical or theoretical reference material.	Little or no evidence of factual information, programming techniques or theoretical explanations being found in technical or theoretical reference material.
Personal and Professional Development Management of learning through self-direction, planning and reflection **REPORT QUALITY 10%**	Outstanding report organisation, structure, presentation, narrative voice and language.	Excellent report organisation, structure, presentation, narrative voice and language.	Very good report organisation, structure, presentation, narrative voice and language.	Good report organisation, structure, presentation, narrative voice and language.	Satisfactory report organisation, structure, presentation, narrative voice and language.	Poor report organisation, structure, presentation, narrative voice and language.	Report does not constitute a serious attempt at the assignment.

How to get help

We will discuss this Coursework Specification in class. However, if you have related questions, please contact Mike Child, childm@lsbu.ac.uk as soon as possible.

Resources

The main resources are the tutorial documents that you used with the Raspberry Pi activities and the activities themselves.

Quality assurance of coursework specifications

Coursework specifications within CSI division go through internal (for new modules with 100% coursework also through external) moderation. This is to ensure high quality, consistency and appropriateness of the coursework as well as to share best practice within the CSI division.

Details of the moderators for this coursework specification are below:

Moderated (internal)	Ioannis Iatropolis, Aarbaz Alam, Panagiotis Alefragkis
Moderated (CSI lead)	[Name, date]
Signed off by (HoD/DHoD)	[Name, date]

----------For Internal use by CSI lead only----------

Changes required to CW?	Yes, No *
Examples of good practice	

*** if changes are required, moderator to complete the below:**

List of changes required	[These needs to be met before signoff can be achieved]
ML Response	[ML response, date]
Moderator Response	[ML response, date]