**UNIVERSITY CENTRE SOMERSET**

**Computing and Digital Technologies**  
Assignment Coversheet and Grading Criteria  
2019 / 2020

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Qualification** | | | **Module Code and Title** | |
| BSc (Hons) Computing and Digital Technologies  FdSc Computing and Digital Technologies | | | SCDT41 Programming and Software Fundamentals | |
| **Student Name and Number** | | | **Module Tutor** | |
|  | | | James Shaun | |
| **Date Issued** | | **Submission Date** | | **Return Date** |
| 17/12/2019 09:00 | | 21/01/2020 15:00 | | 11/01/2020 12:00 |
|  | | |  | |
| **Assignment Number** | 1 of 2. This assignment is worth 50% of the overall module. | | | |
| **Assignment Title** | Coursework One – Programming Portfolio | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Module Learning Outcomes**  *To achieve the outcomes the evidence must show that the learner is able to:* | |  | **Task no.** |
| C1 | Produce programming solutions to solve problems, using high quality code and industry standard practices. |  | 1 - 4 |
| D1 | Apply logical thinking and a creative approach to problem solving. |  | 1 - 4 |

|  |  |
| --- | --- |
| **Word Count of Submission** | Equivalent to 2000 |
| **Student Declaration** | |
| Through submitting this assignment through Turnitin you agree that the work was prepared entirely by yourself in accordance with Open University’s Prevention of Academic Dishonesty Code of Practice. | |

|  |
| --- |
| **Assignment Feedback** |
| All feedback for this assignment will be provided through Turnitin in accordance with the grading criteria below on the given return date. |

|  |  |
| --- | --- |
| **Assignment Task(s)** | |
| **Task no.** | **Task details** |
| Intro | In this assignment, you are required to assemble a portfolio of completed programming solutions to given problems.  A total of four programming solutions need to be completed, documented and submitted.  The evidence of completed work should include:   * A list of requirements for each problem after decomposition * Appropriate screen shots that adequately convey the application’s functionality * Annotated code listings * Evidence of validation if appropriate * Evidence of extra functionality if appropriate   All evidence for the below tasks should be submitted in one portfolio.  In each of the tasks, marks will be award for the following:   * Formal code quality including indentation, brackets, meaningful and helpful comments, spacing of code lines * Correct declarations with appropriate types and meaningful names * Correct user prompts for input and user interface (CLI / GUI) * Correct programming structures and boundary conditions * Correct functionality and calculations * Effective choice of programming data structures and classes. * Well formatted output on screen * Any additional functionality and suitable features * Overall quality of portfolio |
| 1 | **Task 1 (10%)**  **A small local rail service requires a new system to record how many passengers are booked across all carriages on a given journey, it must include the number of pre-booked passengers and the number of passengers that pay on the arrival.**  Create a solution where the values are input into the system then added together and stored. The passenger totals should then be sorted from smallest to largest and output to the console. Your solution should use suitable data structures.  *For example:*  **Pre-booked passenger numbers:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Coach A | Coach B | Coach C | Coach D | Coach E | | 13 | 4 | 8 | 9 | 2 |   **Pay on arrival passenger numbers:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Coach A | Coach B | Coach C | Coach D | Coach E | | 2 | 3 | 57 | 8 | 9 |   **Total number of passengers:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Coach B | Coach E | Coach A | Coach D | Coach C | | 7 | 11 | 15 | 17 | 65 |   **Task 2 (15%)**  **A local primary school requires a spelling quiz for a variety of commonly misspelled words.**  Create a solution which checks a student’s input for the following spelling quiz:   |  |  | | --- | --- | | **Question** | **Answer** | | The second month in the year is | February | | When you have made your mind up, you have | Decided | | If something is not boring, it is | Interesting | | Not all the time but | Sometimes | | Not the same but | Different |   Give 10 points for each correct spelling, deduct 1 mark for each incorrect letter. The solution should output the student’s final mark as a percentage.  **Task 3 (15%)**  **A small independent company requires a simple login system.**  Create a login system that allows users to enter a username and password. The login system must only authenticate access to the following users:   |  |  |  |  | | --- | --- | --- | --- | | Username | Password | Firstname | Surname | | JOHNSM22738 (Admin) | D7y6a | John | Smith | | JANEDO98786 | i&acN | Jane | Doe | | BRYNW56655 | GgjN6 | Bryn | Williams | | NESSAJ25255 | 3KsyX | Nessa | Jenkins |   The solution should check both username and password and output any suitable error messages.  There should be a maximum of three attempts before the user is locked out until they are unlocked by an Admin or the program has been restarted, the number of remaining attempts should be displayed to the user.  On correct entry to the system a welcome message should be displayed to the user e.g. ‘Welcome, Nessa Jenkins’.  The Admin should be able to add and remove users from the system.  **Task 4 (50%)**  **You are required to create an OOP Library Book Loan System.**  The system should allow an Administrator to manage the following:   * Customers * Books * Loans   As a minimum the Administrator should be able to:   * Add and retrieve Customer details; * Add and retrieve and retrieve Book details; * Loan Books to Customers; * Retrieve Loan details; |

|  |
| --- |
| **Sources of Information** |
| Clarke, N. (2017) *C#: A Detailed Approach to Practical Coding*, Create Space Independent  Greene, J. and Stellman, A. (2013) *Head First C#,* O’Reilly Media  Haunts, S. (2017) *A Gentle Introduction to Agile Software Development,* Stephen Haunts Ltd.  McGrath, M. (2017) *C# Programming in Easy Steps*, In Easy Steps Ltd. |

|  |
| --- |
| **Submission Requirements** |
| 1. Check the grading criteria below to ensure your assignment document meets the demands of the above task(s). 2. Include screenshots of each solution with 3. Name the assignment document and any relevant ZIP file using the following format:  SCGT42 \_CW1\_*StudentNumber\_FirstName\_LastName*.docx/zip  (replace the *placeholders* with your student number, first and last name respectively) 4. Go to the Turnitin and use the upload facility to submit your assignment and any required ZIP file to the relevant module. There is no need to submit this assignment brief. |

|  |  |
| --- | --- |
| **IMPORTANT INFORMATION** | |
| * Please stay within the limits of the word count stated at the top of assignment brief. Any additional content over the word count limit (plus or minus 10%) will be disregarded and not be assessed at all.  All work should be submitted online via Turnitin.Please ensure that you submit your assignment on the right submission slot for each module.It is your responsibility to check that you can access Turnitin and Blackboard properly. If your college student account is locked, please contact IT on 01823 366 354 or email them to ITSHelpdesk@btc.ac.uk and request to have your account unlocked, but please ensure you allow plenty of time to do this, do not leave everything until the last day of your deadline.If there are circumstances where you need to submit your assignment other than online, please discuss your needs with the module tutor and alternative arrangements could be made so that you can submit your coursework within the set deadline.Regulations allow you to submit coursework up to 6 working days late. A penalty of deducting 10% will be applied for each day an assignment is late, with a maximum penalty of deducting 60% from your final mark for the late assignment. Any assignment submitted later than 6 days with be awarded a mark of zero. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Numeric Grade | Descriptor  (Class Band) |  | Undergraduate Grading Criteria | | | | |
| Problem 1 - (10%) | | Problem 2- (15%) | Problem 3- (15%) | Problem 4- (50%) | Overall Quality of Portfolio (10%) |
| 80-100 | Outstanding  (Upper Distinction) | Industry standard solution which will meets and exceeds all given requirements. Evidence of wider reading beyond the scope if the module has been demonstrated. Additonal suitable functionality and/or features have been included. Suitable generic data structures have been used such as Lists. | | Industry standard solution which will meets and exceeds all given requirements. Evidence of wider reading beyond the scope if the module has been demonstrated. Additional suitable functionality and/or features have been included. Suitable use of validation, extension of word library and correct calculation and output of results. | Industry standard solution which will meets and exceeds all given requirements. Evidence of wider reading beyond the scope if the module has been demonstrated. Additional suitable functionality and/or features have been included giving users more scope to customise or adjust based on growing needs of the business, such as with users, classes, manipulation of stored passwords, privatisation and encapsulation. | Industry standard solution which will meets and exceeds all given requirements. Evidence of wider reading beyond the scope if the module has been demonstrated. Additional suitable functionality and/or features have been included. Suitable use of classes, instantiation, relationships, efficiency, functionality, validation, collections and inheritance/extension. | Portfolio is of professional quality. All code has been well annotated accurately using technical terminology and confidence. There are no spelling, grammatical or punctuation errors evident. |
| 70-79 | Excellent  (Lower Distinction) | Solution is of excellent quality, with all features expected included in final build alongside many extended features not in original requirements. Good evidence of wider reading or industry standard techniques for software and programming fundamentals. | | Solution is of excellent quality, with all features expected included in final build alongside many extended features not in original requirements. Good evidence of wider reading or industry standard techniques for software and programming fundamentals. | Solution is of excellent quality, with all features expected included in final build alongside many extended features not in original requirements. Good evidence of wider reading or industry standard techniques for software and programming fundamentals. | Solution is of excellent quality, with all features expected included in final build alongside many extended features not in original requirements. Good evidence of wider reading or industry standard techniques for software and programming fundamentals. | Portfolio is of excellent overall quality. All code has been well annotated using technical terminology with minor mistakes. There are very few minor spelling, grammatical or punctuation errors evident. |
| 60-69 | Very Good (Commendation) | Solution is of very good quality and uses effective techniques to achieve the goals of the solution. The solution includes some extended features that improves overall functionality as well as meeting all requirements. The overall solution provides very good demonstration and competence of skills for software and programming fundamentals as well as some evidence of wider reading. | | Solution is of very good quality and uses effective techniques to achieve the goals of the solution. The solution includes some extended features that improves overall functionality as well as meeting all requirements. The overall solution provides very good demonstration and competence of skills for software and programming fundamentals as well as some evidence of wider reading. | Solution is of very good quality and uses effective techniques to achieve the goals of the solution. The solution includes some extended features that improves overall functionality such as additional users, password resets/changes, or private values. The overall solution provides very good demonstration and competence of skills for software and programming fundamentals and meets all requirements. | Solution is of very good quality and uses effective techniques to achieve the goals of the solution. The solution includes some extended features not in the initial brief such as manipulation of existing data, extension of functions protection levels and functionality that improves overall functionality. The overall solution provides very good demonstration and competence of skills and meets all requirements. | Portfolio is of very good overall quality, demonstrates a range of techniques effectively and with few errors. Terminology will be correct and presentation will be of a very good standard. |
| 50-59 | Good/Satisfactory  (Upper Pass) | Solution is of good quality with good levels of functionality which meet most of the solutions requirements, and demonstrates good skills and knowledge of programming and software fundamentals | | Solution is of good quality with good levels of functionality which meet most of the solutions requirements , and demonstrates good skills and knowledge of programming and software fundamentals | Solution is of good quality with good levels of functionality which meet most of the solutions requirements , and demonstrates good skills and knowledge of programming and software fundamentals | Solution is of good quality with good levels of functionality which meet most of the solutions requirements , and demonstrates good skills and knowledge of programming and software fundamentals | Portfolio is of good overall quality, demonstrates a range of techniques clearly and with some minor errors. Terminology will generally be correct and presentation will be of a good standard. |
| 40-49 | Marginal Pass / Satisfactory (Lower Pass) | Solution is of satisfactory quality and demonstrates some of the appropriate skills and knowledge of software and programming fundamentals. The complete solution will meet some of the problems requirements. | | Solution is of satisfactory quality and demonstrates some of the appropriate skills and knowledge of software and programming fundamentals and meets some requirements. | Solution is of satisfactory quality and demonstrates some of the appropriate skills and knowledge of software and programming fundamentals. The complete solution will meet some of the problems requirements. | Solution is of satisfactory quality and demonstrates some of the appropriate skills and knowledge of software and programming fundamentals. The complete solution will meet some of the problems requirements. | Portfolio is of marginally satisfactory quality, includes require evidence of tasks and some annotations of code. There will be many mistakes or inaccuracies. |
| 20-39 | Clear Fail  (Fail) | Solution is of poor quality and shows a lack of skills and knowledge in software and programming fundamentals. | | Solution is of poor quality and shows a lack of skills and knowledge in software and programming fundamentals. | Solution is of poor quality and shows a lack of skills and knowledge in software and programming fundamentals. | Solution is of poor quality and shows a lack of skills and knowledge in software and programming fundamentals. | Portfolio is incomplete, with many missing features or content, not meeting the requirements of tasks, no additions of code or evidence. |
|  | Nothing of Merit  (Fail) | Solution does not meet any of the given requirements and/or nothing of merit has been submitted. | | Solution does not meet any of the given requirements and/or nothing of merit has been submitted. | Solution does not meet any of the given requirements and/or nothing of merit has been submitted. | Solution does not meet any of the given requirements and/or nothing of merit has been submitted. | Portfolio is incomplete and/or has very poor structure. |