|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name: |  | Student ID: |  | Lecturer Code: |  |

|  |
| --- |
| **Course Learning Outcomes** |
| 1. Explain the fundamental of software development and programming concepts (C2, PLO1) |
| 1. Construct a programmable solution using appropriate problem-solving methods and programming concepts to given scenario. (C3, PLO2) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Design  (10%)  CLO1 – PLO2 | Fail | Marginal Fail | Pass | Merit | Distinction |
| 0 | 1 - 2 | 3 - 4 | 5 - 7 | 8 - 10 |
| * Inappropriate or no pseudocode submitted. * Pseudocode covers less than 40% of system requirements with correct logic/solution. | * Pseudocode covers between 40% - 50% of system requirements with correct logic/solution. | * Pseudocode covers between 50% - 65% of system requirements with correct logic/solution. | * Pseudocode covers between 65% - 80% of system requirements with correct logic/solution. | * Pseudocode covers more than 80% of system requirements with correct logic/solution. |
|  |  |  |  |  |
| * Inappropriate or no flowchart submitted. * Flowchart covers less than 40% of system requirements with correct logic/solution and shapes. | * Flowchart covers between 40% - 50% of system requirements with correct logic/solution and shapes. | * Flowchart covers between 50% - 65% of system requirements with correct logic/solution and shapes. | * Flowchart covers between 65% - 80% of system requirements with correct logic/solution and shapes. | * Flowchart covers more than 80% of system requirements with correct logic/solution and shapes. |
|  |  |  |  |  |
|  |  |  |  |  | Total: / 10 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Coding  (Implementation)  (20%)  CLO3 – PLO3 | Fail | Marginal Fail | Pass | Merit | Distinction |
| 0 - 1 | 2 - 3 | 4 - 6 | 7 - 9 | 10 - 12 |
| * No program submitted. * Incomplete / illogical solution. * Program has major errors, does not compile and/or does not run when executed. * Solution/output meets less than 40% of system requirements. * No / least mapping between program design and solution. | * Program compiles with no/some errors and runs when executed. * Solution/output meets between 40% - 50% of system requirements. * Little or no mapping between mapping between program design and solution. | * Program compiles with no/some errors and runs when executed. * Solution/output meets between 50% - 65% of system requirements. * Average/some mapping between program design and solution. | * Program compiles with no/minimum errors and runs when executed. * Solution/output meets between 65% - 80% of system requirements. * Good mapping between program design and solution. | * Program compiles with no errors and runs when executed. * Solution/output meets more than 80% of system requirements. * Excellent mapping between program design and solution. |
|  |  |  |  |  |
| 0 - 3 | 4 | 5 | 6 | 7 - 8 |
| * No program submitted. * Only basic programming techniques applied to build program solution. * Application of programming techniques like modular programming, menu-driven application development, nested lists, nested control structures and File I/O is not evident. | * Program solution shows application of programming techniques like modular programming, menu-driven application development, nested lists, nested control structures and File I/O is minimum or below average level. | * Program solution shows average use of programming techniques like modular programming, menu-driven application development, nested lists, nested control structures and File I/O. | * Program solution shows adequate application of programming techniques like modular programming, menu-driven application development, nested lists, nested control structures and File I/O. | * Program solution shows mastery level of candidate in using programming techniques like modular programming, menu-driven application development, nested lists, nested control structures and File I/O. |
| * No program submitted. * Very poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is less than 40%. * No validation. | * Poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 40% - 50% only. * Poor/minor validation. | * Basic coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 50% - 65% only. * Average/some validation. | * Adherence to good programming practices like commenting, variable naming and indentation is between 65% - 80% only. * Good validation. | * Excellent adherence to good programming practices like commenting, variable naming and indentation. * Excellent validation. |
|  |  |  |  |  |
|  |  |  |  |  | Total: / 20 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Documentation  (10%)  CLO2 – PLO1 | Fail | Marginal Fail | Pass | Credit | Distinction |
| 0 - 1 | 2 - 4 | 5 - 7 | 8 - 9 | 10 |
| * No source code included. * No / poor / inaccurate explanation on the internal working of codes in program solution. | * Major printout of source codes not included. * Insufficient explanation on the internal working of codes in program solution. | * Document missing some minor printout of source codes. * Moderate explanation on the internal working of codes in program solution. | * Most of the source code printout included in the documentation. * Provided detail and accurate explanation on the internal working of codes in program solution. | * All source codes included in documentation. * Provided detail and accurate explanation on the internal working of all codes in program solution. |
| * No documentation submitted or shows no attention to documentation format and structure. * No or very minimum sample input/output screenshots attached in documentation without any explanation. * No text file attached in documentation. | * Shows least attention to documentation format and structure. * Insufficient sample input/output screenshots attached in documentation without or with minimum explanation. * Not all created text files are attached in documentation. | * Shows moderate attention to documentation format and structure. * Moderate amount of sample input/output screenshots attached in documentation with some explanation. * Almost all created text files are attached in documentation. | * Shows sufficient level of attention to documentation format and structure. * Sample input/output screenshots attached in documentation is almost comprehensive with adequate explanation. * All created text files are attached in documentation. | * Shows high degree of attention to documentation format and structure. * Sample input/output screenshots attached in documentation is comprehensive and explained in detail. * All created text files are attached in documentation. |
|  |  |  |  |  |
|  |  |  |  |  | Total: / 10 |
| Demonstration/ Presentation  (10%)  CLO2 – PLO1 | 0 - 1 | 2 - 3 | 4 - 5 | 6 - 7 | 8 - 10 |
| * Did not turn up for presentation. * Not able to trace any of the codes / work done. * Unable or barely able to answer any of the question asked. | * Barely able to trace the codes / work done. * Mostly inaccurate / illogical answers / explanation provided or barely able to answer some of the questions asked | * Able to trace some codes / work done with hesitation. * Able to answer some questions posed accurately or logically. | * Able to trace the codes and work done. * Able to answer most questions posed accurately and shows a good understanding of how the program works. | * In depth understanding of the codes / work done. * Able to answer all questions posed with minimal omissions. * Show additional concepts / new ideas used in the solution. |
|  |  |  |  |  |
|  |  |  |  |  | Total: / 10 |