Marking Scheme

Student 1 : **TP Number :**

	Fail	Marginal Fail	Pass	Merit	Distinction
	0 - 9	**10 - 14**	**15 -18**	**19 - 23**	**24 - 300**
Design (30%)	• Inappropriate or no pseudocode/flowchart submitted. • Pseudocode/flowchart covers less than 40% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 40% - 50% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 50% - 65% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 65% - 80% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers more than 80% of system requirements with correct logic/solution.
	Fail	**Marginal Fail**	**Pass**	**Merit**	**Distinction**
	0 - 13	**14 - 19**	**20 -25**	**26 - 31**	**32 - 40**
Coding (Implementation) (40%)	• 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 solution shows application of programming techniques like modular	• 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 solution shows average use of programming techniques like modular programming, menu-	• 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 solution shows adequate application of programming techniques like modular programming, menu-	• 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. • Program solution shows mastery level of candidate in using programming techniques like modular programming, menu-driven application development,

	• 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 or data storage is not evident. • Very poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is less than 40%. No validation.	programming, menu-driven application development, nested lists, nested control structures and File I/O or data storage is minimum or below average level. • Poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 40% - 50% only. • Poor/minor validation.	driven application development, nested lists, nested control structures and File I/O or data storage. • Basic coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 50% - 65% only. • Average/some validation.	driven application development, nested lists, nested control structures and File I/O or data storage. • Adherence to good programming practices like commenting, variable naming and indentation is between 65% - 80% only. • Good validation.	nested lists, nested control structures and File I/O or data storage. • Excellent adherence to good programming practices like commenting, variable naming and indentation. Excellent validation.
Documentation (20%)	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0 - 6	**7 - 9**	**10 -12**	**13 - 15**	**16 - 20**
	• No documentation submitted or shows no attention to documentation format and structure. • No or very minimum sample input/output screenshots attached in	• Shows least attention to documentation format and structure. • Insufficient sample input/output screenshots attached in documentation	• Shows moderate attention to documentation format and structure. • Moderate amount of sample input/output screenshots attached in	• Shows sufficient level of attention to documentation format and structure. • Sample input/output screenshots attached in documentation is almost	• Shows high degree of attention to documentation format and structure. • Sample input/output screenshots attached in documentation is

	documentation without any explanation. • No code snippet included. No / poor / inaccurate explanation on the internal working of codes in program solution.	without or with minimum explanation. • Major printout of code snippet not included. Insufficient explanation on the internal working of codes in program solution.	documentation with some explanation. • Document missing some minor printout of code snippet. Moderate explanation on the internal working of codes in program solution.	comprehensive with adequate explanation. • Most of the source code snippet included in the documentation. Provided detail and accurate explanation on the internal working of codes in program solution.	comprehensive and explained in detail. • All code snippet included in documentation. Provided detail and accurate explanation on the internal working of all codes in program solution.
Presentation (10%)	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0 - 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.

Student 2 : **TP Number :**

	Fail	Marginal Fail	Pass	Merit	Distinction
	0 - 9	**10 - 14**	**15 -18**	**19 - 23**	**24 - 300**
Design (30%)	• Inappropriate or no pseudocode/flowchart submitted. • Pseudocode/flowchart covers less than 40% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 40% - 50% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 50% - 65% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 65% - 80% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers more than 80% of system requirements with correct logic/solution.
	Fail	**Marginal Fail**	**Pass**	**Merit**	**Distinction**
	0 - 13	**14 - 19**	**20 -25**	**26 - 31**	**32 - 40**
Coding (Implementation) (40%)	• 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. • Only basic programming techniques applied to build program 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 solution shows application of programming techniques like modular programming, menu-driven application	• 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 solution shows average use of programming techniques like modular programming, menu-driven application development, nested lists,	• 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 solution shows adequate application of programming techniques like modular programming, menu-driven application development, nested lists,	• 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. • Program solution shows mastery level of candidate in using programming techniques like modular programming, menu-driven application development, nested lists, nested control

	• Application of programming techniques like modular programming, menu-driven application development, nested lists, nested control structures and File I/O or data storage is not evident. • Very poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is less than 40%. No validation.	development, nested lists, nested control structures and File I/O or data storage is minimum or below average level. • Poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 40% - 50% only. • Poor/minor validation.	nested control structures and File I/O or data storage. • Basic coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 50% - 65% only. • Average/some validation.	nested control structures and File I/O or data storage. • Adherence to good programming practices like commenting, variable naming and indentation is between 65% - 80% only. • Good validation.	structures and File I/O or data storage. • Excellent adherence to good programming practices like commenting, variable naming and indentation. Excellent validation.
Documentation (20%)	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0 - 6	**7 - 9**	**10 -12**	**13 - 15**	**16 - 20**
	• 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 code snippet included.	• Shows least attention to documentation format and structure. • Insufficient sample input/output screenshots attached in documentation without or with minimum explanation. • Major printout of code snippet not included.	• Shows moderate attention to documentation format and structure. • Moderate amount of sample input/output screenshots attached in documentation with some explanation.	• Shows sufficient level of attention to documentation format and structure. • Sample input/output screenshots attached in documentation is almost comprehensive with adequate explanation.	• Shows high degree of attention to documentation format and structure. • Sample input/output screenshots attached in documentation is comprehensive and explained in detail. • All code snippet included in documentation.

	No / poor / inaccurate explanation on the internal working of codes in program solution.	Insufficient explanation on the internal working of codes in program solution.	• Document missing some minor printout of code snippet. Moderate explanation on the internal working of codes in program solution.	• Most of the source code snippet included in the documentation. Provided detail and accurate explanation on the internal working of codes in program solution.	Provided detail and accurate explanation on the internal working of all codes in program solution.
Presentation (10%)	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0 - 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.

Student 3 : **TP Number :**

	Fail	Marginal Fail	Pass	Merit	Distinction
	0 - 9	**10 - 14**	**15 -18**	**19 - 23**	**24 - 300**
Design (30%)	• Inappropriate or no pseudocode/flowchart submitted. • Pseudocode/flowchart covers less than 40% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 40% - 50% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 50% - 65% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 65% - 80% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers more than 80% of system requirements with correct logic/solution.
	Fail	**Marginal Fail**	**Pass**	**Merit**	**Distinction**
	0 - 13	**14 - 19**	**20 -25**	**26 - 31**	**32 - 40**
Coding (Implementation) (40%)	• 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. • Only basic programming techniques applied to build program 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 solution shows application of programming techniques like modular programming, menu-driven application	• 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 solution shows average use of programming techniques like modular programming, menu-driven application development, nested lists,	• 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 solution shows adequate application of programming techniques like modular programming, menu-driven application development, nested lists,	• 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. • Program solution shows mastery level of candidate in using programming techniques like modular programming, menu-driven application development, nested lists, nested control

	• Application of programming techniques like modular programming, menu-driven application development, nested lists, nested control structures and File I/O or data storage is not evident. • Very poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is less than 40%. No validation.	development, nested lists, nested control structures and File I/O or data storage is minimum or below average level. • Poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 40% - 50% only. • Poor/minor validation.	nested control structures and File I/O or data storage. • Basic coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 50% - 65% only. • Average/some validation.	nested control structures and File I/O or data storage. • Adherence to good programming practices like commenting, variable naming and indentation is between 65% - 80% only. • Good validation.	structures and File I/O or data storage. • Excellent adherence to good programming practices like commenting, variable naming and indentation. Excellent validation.
Documentation (20%)	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0 - 6	**7 - 9**	**10 -12**	**13 - 15**	**16 - 20**
	• 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 code snippet included.	• Shows least attention to documentation format and structure. • Insufficient sample input/output screenshots attached in documentation without or with minimum explanation. • Major printout of code snippet not included.	• Shows moderate attention to documentation format and structure. • Moderate amount of sample input/output screenshots attached in documentation with some explanation.	• Shows sufficient level of attention to documentation format and structure. • Sample input/output screenshots attached in documentation is almost comprehensive with adequate explanation.	• Shows high degree of attention to documentation format and structure. • Sample input/output screenshots attached in documentation is comprehensive and explained in detail. • All code snippet included in documentation.

	No / poor / inaccurate explanation on the internal working of codes in program solution.	Insufficient explanation on the internal working of codes in program solution.	• Document missing some minor printout of code snippet. Moderate explanation on the internal working of codes in program solution.	• Most of the source code snippet included in the documentation. Provided detail and accurate explanation on the internal working of codes in program solution.	Provided detail and accurate explanation on the internal working of all codes in program solution.
Presentation (10%)	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0 - 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.

Student 4 : **TP Number :**

	Fail	Marginal Fail	Pass	Merit	Distinction
	0 - 9	**10 - 14**	**15 -18**	**19 - 23**	**24 - 300**
Design (30%)	• Inappropriate or no pseudocode/flowchart submitted. • Pseudocode/flowchart covers less than 40% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 40% - 50% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 50% - 65% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers between 65% - 80% of system requirements with correct logic/solution.	• Pseudocode/flowchart covers more than 80% of system requirements with correct logic/solution.
	Fail	**Marginal Fail**	**Pass**	**Merit**	**Distinction**
	0 - 13	**14 - 19**	**20 -25**	**26 - 31**	**32 - 40**
Coding (Implementation) (40%)	• 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. • Only basic programming techniques applied to build program 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 solution shows application of programming techniques like modular programming, menu-driven application	• 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 solution shows average use of programming techniques like modular programming, menu-driven application development, nested lists,	• 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 solution shows adequate application of programming techniques like modular programming, menu-driven application development, nested lists,	• 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. • Program solution shows mastery level of candidate in using programming techniques like modular programming, menu-driven application development, nested lists, nested control

	• Application of programming techniques like modular programming, menu-driven application development, nested lists, nested control structures and File I/O or data storage is not evident. • Very poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is less than 40%. No validation.	development, nested lists, nested control structures and File I/O or data storage is minimum or below average level. • Poor coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 40% - 50% only. • Poor/minor validation.	nested control structures and File I/O or data storage. • Basic coding style. Adherence to good programming practices like commenting, variable naming and indentation is between 50% - 65% only. • Average/some validation.	nested control structures and File I/O or data storage. • Adherence to good programming practices like commenting, variable naming and indentation is between 65% - 80% only. • Good validation.	structures and File I/O or data storage. • Excellent adherence to good programming practices like commenting, variable naming and indentation. Excellent validation.
Documentation (20%)	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0 - 6	**7 - 9**	**10 -12**	**13 - 15**	**16 - 20**
	• 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 code snippet included.	• Shows least attention to documentation format and structure. • Insufficient sample input/output screenshots attached in documentation without or with minimum explanation. • Major printout of code snippet not included.	• Shows moderate attention to documentation format and structure. • Moderate amount of sample input/output screenshots attached in documentation with some explanation.	• Shows sufficient level of attention to documentation format and structure. • Sample input/output screenshots attached in documentation is almost comprehensive with adequate explanation.	• Shows high degree of attention to documentation format and structure. • Sample input/output screenshots attached in documentation is comprehensive and explained in detail. • All code snippet included in documentation.

	No / poor / inaccurate explanation on the internal working of codes in program solution.	Insufficient explanation on the internal working of codes in program solution.	• Document missing some minor printout of code snippet. Moderate explanation on the internal working of codes in program solution.	• Most of the source code snippet included in the documentation. Provided detail and accurate explanation on the internal working of codes in program solution.	Provided detail and accurate explanation on the internal working of all codes in program solution.
Presentation (10%)	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0 - 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.