CT108-3-1 Python Programming Page 1 of 12

Marking Scheme

Student 1:

	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
	No program submitted.	Program compiles with no/some errors and runs	Program compiles with no/some errors and runs	Program compiles with no/minimum errors and	Program compiles with no errors and runs when
	• Incomplete / illogical solution.	when executed.	when executed.	runs when executed.	errors and runs when executed.
Coding	• Program has major errors,	• Solution/output meets	• Solution/output meets	• Solution/output meets	• Solution/output meets more
(Implementation) (40%)	does not compile and/or does not run when	between 40% - 50% of system requirements.	between 50% - 65% of system requirements.	between 65% - 80% of system requirements.	than 80% of system requirements.
(40 /0)	executed.	• Little or no mapping	• Average/some mapping	• Good mapping between	• Excellent mapping between
	• Solution/output meets less	between mapping between	between program design and solution.	program design and solution.	program design and solution.
	than 40% of system requirements.	program design and solution.	Program solution shows	• Program solution shows	• Program solution shows mastery level of candidate in
	• No / least mapping	• Program solution shows	average use of	adequate application of	using programming
	between program design	application of	programming techniques	programming techniques	techniques like modular
	and solution.	programming techniques like modular	like modular programming, menu-	like modular programming, menu-	programming, menu-driven application development,

TP Number:

CT108-3-1 Python Programming Page 2 of 12

CT108-3-1 Python Programming Page 3 of 12

	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.	minor printout of code snippet.	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.
	Fail 0 - 3	Marginal Fail 4	Pass 5 - 6	Credit 7	Distinction 8 - 10
Presentation (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.

CT108-3-1 Python Programming Page 4 of 12

Student 2: TP Number:

	Fail	Marginal Fail	Pass	Merit	Distinction
		8			
	0 - 9	10 - 14	15 -18	19 - 23	24 - 300
Design	• Inappropriate or no pseudocode/flowchart	• Pseudocode/flowchart covers between 40% -	• Pseudocode/flowchart covers between 50% -	• Pseudocode/flowchart covers between 65% -	Pseudocode/flowchart covers more than 80% of system
_	submitted.	50% of system	65% of system	80% of system	requirements with correct
(30%)	Pseudocode/flowchart	requirements with correct	requirements with correct	requirements with correct	logic/solution.
	covers less than 40% of	logic/solution.	logic/solution.	logic/solution.	
	system requirements with				
	correct logic/solution.				
	Fail	Marginal Fail	Pass	Merit	Distinction
	0 - 13	14 - 19	20 -25	26 - 31	32 - 40
	No program submitted.	• Program compiles with	• Program compiles with	• Program compiles with	• Program compiles with no
	• Incomplete / illogical solution.	no/some errors and runs when executed.	no/some errors and runs when executed.	no/minimum errors and runs when executed.	errors and runs when executed.
	• Program has major errors,	• Solution/output meets	• Solution/output meets	• Solution/output meets	• Solution/output meets more
Coding	does not compile and/or	between 40% - 50% of	between 50% - 65% of	between 65% - 80% of	than 80% of system
(Implementation)	does not run when	system requirements.	system requirements.	system requirements.	requirements.
(40%)	executed.	• Little or no mapping	• Average/some mapping	• Good mapping between	• Excellent mapping between
, ,	• Solution/output meets less	between mapping between	between program design	program design and	program design and solution.
	than 40% of system	program design and	and solution.	solution.	• Program solution shows
	requirements.	solution.	• Program solution shows	• Program solution shows	mastery level of candidate in
	• No / least mapping	• Program solution shows	average use of	adequate application of	using programming
	between program design	application of	programming techniques	programming techniques	techniques like modular
	and solution.	programming techniques	like modular	like modular	programming, menu-driven
	• Only basic programming	like modular	programming, menu-	programming, menu-	application development,
	techniques applied to build	programming, menu- driven application	driven application development, nested lists,	driven application development, nested lists,	nested lists, nested control
	program solution.	driven application	development, nested lists,	development, nested lists,	

CT108-3-1 Python Programming Page 5 of 12

	• Application of	development, nested lists,	nested control structures	nested control structures	structures and File I/O or data
	programming techniques	nested control structures	and File I/O or data	and File I/O or data	storage.
	like modular	and File I/O or data	storage.	storage.	• Excellent adherence to good
	programming, menu-	storage is minimum or	• Basic coding style.	• Adherence to good	programming practices like
	driven application	below average level.	Adherence to good	programming practices	commenting, variable naming
	development, nested lists,	• Poor coding style.	programming practices	like commenting, variable	and indentation.
	nested control structures	Adherence to good	like commenting, variable	naming and indentation is	Excellent validation.
	and File I/O or data	programming practices	naming and indentation is	between 65% - 80% only.	
	storage is not evident.	like commenting, variable	between 50% - 65% only.	Good validation.	
	• Very poor coding style.	naming and indentation is	Average/some validation.		
	Adherence to good	between 40% - 50% only.			
	programming practices	• Poor/minor validation.			
	like commenting, variable				
	naming and indentation is				
	less than 40%.				
	No validation.				
	Fail	Marginal Fail	Pass	Credit	Distinction
	0 - 6	7 - 9	10 -12	13 - 15	16 - 20
Documentation	No documentation	• Shows least attention to	• Shows moderate attention	• Shows sufficient level of	• Shows high degree of attention
(20%)	submitted or shows no	documentation format and	to documentation format	attention to documentation	to documentation format and
(2070)	attention to documentation	structure.	and structure.	format and structure.	structure.
	format and structure.	• Insufficient sample	• Moderate amount of	• Sample input/output	• Sample input/output
	• No or very minimum	input/output screenshots	sample input/output	screenshots attached in	screenshots attached in
	sample input/output screenshots attached in	attached in documentation	screenshots attached in documentation with some	documentation is almost	documentation is
	documentation without	without or with minimum explanation.	explanation.	comprehensive with adequate explanation.	comprehensive and explained in detail.
	any explanation.	Major printout of code	CAPIAHAHOH.	aucquate explanation.	All code snippet included in
	any explanation.	viajoi printout of code			- An code simpler included in
	• No code snippet included.	snippet not included.			documentation.

CT108-3-1 Python Programming Page 6 of 12

	No / poor / inaccurate	Insufficient explanation on	• Document missing some	• Most of the source code	Provided detail and accurate
	explanation on the internal	the internal working of	minor printout of code	snippet included in the	explanation on the internal
	working of codes in	codes in program solution.	snippet.	documentation.	working of all codes in
	program solution.		Moderate explanation on	Provided detail and	program solution.
			the internal working of	accurate explanation on	
			codes in program solution.	the internal working of	
				codes in program solution.	
	Fail	Marginal Fail	Pass	Credit	Distinction
	0 - 3	4	5 - 6	7	0 10
	0-3	-	5-0	1	8 - 10
	0-3	*	5-0	,	8 - 10
D	• Did not turn up for	Barely able to trace the	• Able to trace some codes /	Able to trace the codes and	• In depth understanding of the
Presentation		-			
Presentation (10%)	• Did not turn up for	Barely able to trace the	• Able to trace some codes /	Able to trace the codes and	• In depth understanding of the
	• Did not turn up for presentation.	Barely able to trace the codes / work done.	Able to trace some codes / work done with hesitation.	Able to trace the codes and work done.	• In depth understanding of the codes / work done.
	Did not turn up for presentation.Not able to trace any of the	Barely able to trace the codes / work done. Mostly inaccurate /	Able to trace some codes / work done with hesitation. Able to answer some	Able to trace the codes and work done. Able to answer most	 In depth understanding of the codes / work done. Able to answer all questions
	 Did not turn up for presentation. Not able to trace any of the codes / work done. 	Barely able to trace the codes / work done. Mostly inaccurate / illogical answers /	Able to trace some codes / work done with hesitation. Able to answer some questions posed accurately	Able to trace the codes and work done. Able to answer most questions posed accurately	 In depth understanding of the codes / work done. Able to answer all questions posed with minimal
	 Did not turn up for presentation. Not able to trace any of the codes / work done. Unable or barely able to 	Barely able to trace the codes / work done. Mostly inaccurate / illogical answers / explanation provided or	Able to trace some codes / work done with hesitation. Able to answer some questions posed accurately	Able to trace the codes and work done. Able to answer most questions posed accurately and shows a good	 In depth understanding of the codes / work done. Able to answer all questions posed with minimal omissions.
	 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 	Barely able to trace the codes / work done. Mostly inaccurate / illogical answers / explanation provided or barely able to answer some	Able to trace some codes / work done with hesitation. Able to answer some questions posed accurately	Able to trace the codes and work done. Able to answer most questions posed accurately and shows a good understanding of how the	 In depth understanding of the codes / work done. Able to answer all questions posed with minimal omissions. Show additional concepts /

CT108-3-1 Python Programming Page 7 of 12

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 mapping between program design and solution. Program solution shows application of programming techniques like modular programming, menudriven 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, menudriven 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, menudriven 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

CT108-3-1 Python Programming Page 8 of 12

			T	T	
	• Application of	development, nested lists,	nested control structures	nested control structures	structures and File I/O or data
	programming techniques	nested control structures	and File I/O or data	and File I/O or data	storage.
	like modular	and File I/O or data	storage.	storage.	• Excellent adherence to good
	programming, menu-	storage is minimum or	• Basic coding style.	• Adherence to good	programming practices like
	driven application	below average level.	Adherence to good	programming practices	commenting, variable naming
	development, nested lists,	• Poor coding style.	programming practices	like commenting, variable	and indentation.
	nested control structures	Adherence to good	like commenting, variable	naming and indentation is	Excellent validation.
	and File I/O or data	programming practices	naming and indentation is	between 65% - 80% only.	
	storage is not evident.	like commenting, variable	between 50% - 65% only.	Good validation.	
	• Very poor coding style.	naming and indentation is	• Average/some validation.	- Good varidation.	
	Adherence to good	between 40% - 50% only.	Average/some vandation.		
	programming practices	• Poor/minor validation.			
		Foor/minor varidation.			
	like commenting, variable				
	naming and indentation is less than 40%.				
	No validation.				
	No vandation.				
	Fail	Marginal Fail	Pass	Credit	Distinction
	0 - 6	7 - 9	10 -12	13 - 15	16 - 20
_	No documentation	• Shows least attention to	• Shows moderate attention	• Shows sufficient level of	Shows high degree of attention
Documentation	submitted or shows no	documentation format and	to documentation format	attention to documentation	to documentation format and
(20%)	attention to documentation	structure.	and structure.	format and structure.	structure.
	format and structure.	• Insufficient sample	• Moderate amount of	• Sample input/output	• Sample input/output
	• No or very minimum	input/output screenshots	sample input/output	screenshots attached in	screenshots attached in
	sample input/output	attached in documentation	screenshots attached in	documentation is almost	documentation is
	screenshots attached in	without or with minimum	documentation with some	comprehensive with	comprehensive and explained
	documentation without	explanation.	explanation.	adequate explanation.	in detail.
	any explanation.	• Major printout of code		and quite explanation.	• All code snippet included in
	 No code snippet included. 	snippet not included.			documentation.
1	- 110 code simplet metaded.	simplet not included.			documentation.

CT108-3-1 Python Programming Page 9 of 12

	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	Most of the source code snippet included in the documentation. Provided detail and accurate explanation on	Provided detail and accurate explanation on the internal working of all codes in program solution.
			codes in program solution.	the internal working of codes in program solution.	
	Fail	Marginal Fail	Pass	Credit	Distinction
	0 - 3	4	5 - 6	7	8 - 10
Presentation (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.

CT108-3-1 Python Programming Page 10 of 12

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.
	7.0			7.7	
	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 compiles with no/some errors and runs when executed. Solution/output meets between 40% - 50% of system requirements. Little or no mapping between program design and solution. Program solution shows application of programming techniques like modular programming, menu- 	 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, menudriven application 	 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, menudriven application 	 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

CT108-3-1 Python Programming Page 11 of 12

			T	T	
	• Application of	development, nested lists,	nested control structures	nested control structures	structures and File I/O or data
	programming techniques	nested control structures	and File I/O or data	and File I/O or data	storage.
	like modular	and File I/O or data	storage.	storage.	• Excellent adherence to good
	programming, menu-	storage is minimum or	• Basic coding style.	• Adherence to good	programming practices like
	driven application	below average level.	Adherence to good	programming practices	commenting, variable naming
	development, nested lists,	• Poor coding style.	programming practices	like commenting, variable	and indentation.
	nested control structures	Adherence to good	like commenting, variable	naming and indentation is	Excellent validation.
	and File I/O or data	programming practices	naming and indentation is	between 65% - 80% only.	
	storage is not evident.	like commenting, variable	between 50% - 65% only.	Good validation.	
	• Very poor coding style.	naming and indentation is	• Average/some validation.	- Good varidation.	
	Adherence to good	between 40% - 50% only.	Average/some vandation.		
	programming practices	• Poor/minor validation.			
		Foor/minor varidation.			
	like commenting, variable				
	naming and indentation is less than 40%.				
	No validation.				
	No vandation.				
	Fail	Marginal Fail	Pass	Credit	Distinction
	0 - 6	7 - 9	10 -12	13 - 15	16 - 20
_	No documentation	• Shows least attention to	• Shows moderate attention	• Shows sufficient level of	Shows high degree of attention
Documentation	submitted or shows no	documentation format and	to documentation format	attention to documentation	to documentation format and
(20%)	attention to documentation	structure.	and structure.	format and structure.	structure.
	format and structure.	• Insufficient sample	• Moderate amount of	• Sample input/output	• Sample input/output
	• No or very minimum	input/output screenshots	sample input/output	screenshots attached in	screenshots attached in
	sample input/output	attached in documentation	screenshots attached in	documentation is almost	documentation is
	screenshots attached in	without or with minimum	documentation with some	comprehensive with	comprehensive and explained
	documentation without	explanation.	explanation.	adequate explanation.	in detail.
	any explanation.	• Major printout of code		and quite explanation.	• All code snippet included in
	 No code snippet included. 	snippet not included.			documentation.
1	- 110 code simplet metaded.	simplet not included.			documentation.

CT108-3-1 Python Programming Page 12 of 12

	No / poor / inaccurate	Insufficient explanation on	• Document missing some	• Most of the source code	Provided detail and accurate
	explanation on the internal	the internal working of	minor printout of code	snippet included in the	explanation on the internal
	working of codes in	codes in program solution.	snippet.	documentation.	working of all codes in
	program solution.		Moderate explanation on	Provided detail and	program solution.
			the internal working of	accurate explanation on	
			codes in program solution.	the internal working of	
				codes in program solution.	
	Fail	Marginal Fail	Pass	Credit	Distinction
	0 - 3	4	5 - 6	7	8 - 10
	0 - 3	7	3-0	,	0 - 10
	0 - 3	7	3-0	,	8 - 10
Dunantation	• Did not turn up for	• Barely able to trace the	• Able to trace some codes /	Able to trace the codes and	• In depth understanding of the
Presentation		-			
Presentation (10%)	• Did not turn up for	Barely able to trace the	Able to trace some codes /	Able to trace the codes and	• In depth understanding of the
	• Did not turn up for presentation.	Barely able to trace the codes / work done.	Able to trace some codes / work done with hesitation.	Able to trace the codes and work done.	• In depth understanding of the codes / work done.
	Did not turn up for presentation.Not able to trace any of the	Barely able to trace the codes / work done. Mostly inaccurate /	Able to trace some codes / work done with hesitation. Able to answer some	Able to trace the codes and work done. Able to answer most	 In depth understanding of the codes / work done. Able to answer all questions
	 Did not turn up for presentation. Not able to trace any of the codes / work done. 	Barely able to trace the codes / work done. Mostly inaccurate / illogical answers /	Able to trace some codes / work done with hesitation. Able to answer some questions posed accurately	Able to trace the codes and work done. Able to answer most questions posed accurately	 In depth understanding of the codes / work done. Able to answer all questions posed with minimal
	 Did not turn up for presentation. Not able to trace any of the codes / work done. Unable or barely able to 	Barely able to trace the codes / work done. Mostly inaccurate / illogical answers / explanation provided or	Able to trace some codes / work done with hesitation. Able to answer some questions posed accurately	Able to trace the codes and work done. Able to answer most questions posed accurately and shows a good	 In depth understanding of the codes / work done. Able to answer all questions posed with minimal omissions.
	 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 	Barely able to trace the codes / work done. Mostly inaccurate / illogical answers / explanation provided or barely able to answer some	Able to trace some codes / work done with hesitation. Able to answer some questions posed accurately	Able to trace the codes and work done. Able to answer most questions posed accurately and shows a good understanding of how the	 In depth understanding of the codes / work done. Able to answer all questions posed with minimal omissions. Show additional concepts /