31125301	Actual Weight (%)	N (0-4.9)	P (5 - 5.9)	C (6 - 6.9)	D (7 - 7.9)	HD (8 - 10)
		Poor **Fails to identify what is required**	**Basic** Reflects the beginnings of understanding what is required	**Expected** Basic understanding and delivery of what is required.	**Good** Reflects a mastery of what is required	**Excellent** Reflects the highest level of performance, beyond what is required
Coding architecture, style and documentation						
1.1 Code architecture	15%	No code Or - Usage of variables have major issues (e.g. Use hard code data without use variables), Completely Mix up the Usage of local variables and global variables - Solution is very inefficient and not optimised - Used external libraries that is not required (Math is fine, the rest is not) - Data type declaration is wrong, make all variables as string data type	- Usage of variables have some major issues (e.g. Use hard code data without variables), Mix up the Usage of local variables and global variables - Some use of algorithm are not efficient and it is not optimised (e.g Repeating code for many cases instead of a systematic approach) - Used external libraries that is not required (Math is fine, the rest is not) - Some data type declaration is wrong, inappropriate of string data type - Poor Usage of control, use of loops has some major issue or very inefficient	- Usage of variables does not have major issues, declare some of variables at the place where it is necessary. Mix up the Usage of local variables and global variables - Adequate Usage of algorithm, solution is efficient - No unnecessary external libraries (Which auto marker will not able to run). math is fine. - Some data type usage are not correct - Adequate Usage of control, use of loops has some issue or inefficient	- Good Usage of variables, declare majority of variables at the place where it is necessary. Can differentiate the Usage of local variables and global variables at some extend - Good Usage of algorithm, solution is efficient and some are optimised - No unnecessary external libraries (Which auto marker will not able to run). math is fine - Good choices of data types, there are very little mistakes of using data types - Good Usage of control, use of loops has slight issue or it is slightly inefficient	- Excellent Usage of variables, declare variables at the place where it is necessary. Can clearly differentiate the Usage of local variables and global variables - Excellent Usage of algorithm, solution is very efficient and optimised - No unnecessary external libraries (Which auto marker will not able to run). math is fine. - Excellent choices of data types, there are no wrong usage of data types - Excellent Usage of control, appropriate use of loops
1.2 Code logic & style	15%	- Logic expression is not very clear, there are ambiguity inside - Naming of variables have issues, inconsistent variable naming style and ambiguous naming - Some code is not very readable (e.g Very long and length logic which exceed screen, or inconsistent indentation) - Little of the code compliance with specification	- Logic expression is clear, little ambiguity - Naming of variables is very basic, not very consistent of using variable naming style and some ambiguous naming - Some code is readable - Few of the code compliance with specification	- Logic expression is clear, little ambiguity - Good naming of variables, some consistency of using variable naming style and some ambiguous naming - Some code is readable - Most of the code compliance with specification	- Logic expression is mostly clear, very little ambiguity - Good naming of variables, little consistency of using variable naming style and little ambiguous naming - Coding is mostly readable - Majority of the code compliance with specification	- Logic expression is clear without ambiguity - Excellent naming of variables, variable naming is very consistent - Coding is readable with no issues or difficulty - Fully compliance with specification
1.3 Documentation	20%	No comments Or - Writing of documentation is poor, Usage of English has lots of grammar mistakes - Provide little to none documentation at the place where it is necessary - Not formatting of comment	- Writing of documentation is poor, Usage of English is has grammar mistakes - Provide little documentation at the place where it is necessary - Little formatting of comments	- Provides writing of documentation, Usage of English is adequate with some grammar mistakes - Provide some documentation at the place where it is necessary - Adequate formatting of comments	- Good writing of documentation, Good Usage of English with little grammar mistakes - Provide majority of documentation at the place where it is necessary - Good formatting of comments	- Excellent writing of documentation, excellent Usage of English without grammar mistakes - Appropriate documentation at the place where it is necessary - Excellent formatting of comments (Appropriate Usage of single line and multi-line comment and appropriate description for files
Working program functionality (details below)						
2.1 Working program functionality	50%	N	P	C	D	HD
Late Penalty 20% Per Day		0 Day	1 Day	2 Day	3 Day	4 Day
Final Total marks	**30.00**					

General comments	

Working program functionality details

	Marks Received	Feedback
Task 1		
Person constructor and getter methods [1 mark]	1	Correct implementation of Person object constructor and get_name()
Person's add_friend and get_friend() [2 marks]	2	Correct implementation of add_friend(person_instance) and get_friends() methods
load_people() [2 marks]	2	Correct implementation of load_people()'s Person object list
Task 2		
Patient's is_contagious method [1 mark]	1	Correct implementation of Patient.is_contagious()
Patient's infect(viral_load) method [2 marks]	2	Correct implementation of Patient.infect(viral_load)
run_simulation() [2 marks]	2	Correct implementation of run_simulation()
Task 3		
x-y axis label (days, counts) [1 mark]	1	Correct x-y axis labeling
user's input (3 values interactive) [2 marks]	2	Correct inputing of user inputs for num days, meeting prob and patient zero hp
visulization produced/saved [2 marks]	2	Visualisation of the run-simulation correctly genererated

Simulation repeat test

This plot depicts your results from
when the simulation is run
(and repeated up to 5 times) in blue,
and the expected results in
the red-shaded area.

The red-shaded area is created by
running the simulation 1000 times.
The black dashed-lines are the
individual simulation results.
The red-line represents the
average of the 1000 simulations.

If 1 out of 5 of your simulation runs
fall inside the red-shaded region,
then your simulation passes this test.
The marks and feedback for
this are displayed in
Task 2's run_sim() section.