Description of the C++ programming developed system to be applied to the fictitious case study company/institution/ enterprise

In this AI-enabled booming era, to be cost-effective, regular waste management system can be augmented to a really optimized, cost-effective and/or greedy Waste Management System when necessary. Based on optimization, cost effective and/or greedy algorithms, this system is important for any waste or garbage collection place in Malaysia. Fully develop a Waste Management System for a privatized but quite large-scaled company or enterprise in Malaysia using C++ programming. All the details regarding this system are provided in the coursework specification which has been documented quite meticulously as below. All the intended submission requirements, format and deliverable(s) of this coursework are as follows:

All program files in (i) working C++ source code file (.cpp file), (ii) object file (.o file), (iii) executable file (.exe file/application file), and/or relevant files involved.

Tasks to be completed in this coursework are indicated in full details as below: 1. Your proposed ideas and concepts in C++ program to develop the Waste Management System, maintain and enhance the system, keep Page 2 it sustained using approaches of Optimized, and/or Non-optimized (i.e. Regular), Minimum Spanning Tree and/or Traveling Salesman Problem and/or Greedy Routes. The suggested features in C++ programming (which are listed as below) to achieve the goal and targets (i.e. to ensure efficiency, cost-effectiveness and/or the maximum efficiency for waste collector using the system). In the system that you are going to develop, you may choose to include Two, Three, Four or all Five approaches of Optimized, and/or Non-optimized (i.e. Regular), and/or Traveling Salesman Problem and/or Minimum Spanning Tree and/or Greedy Routes. - User-defined data types; libraries involved - Inheritance; Polymorphism; File I/O; Exception Handling - Arrays; Pointers; Array of Pointers; Linked Lists; Vectors - Structures and/or other data structures such as stacks, queues, or the like - Loops such as for-do loops, while-do loops, do-while loops and the like - Control statements such as if-else, if-then-else, nested if, and the like - Random number generator (for random garbage/waste level generation) - Other relevant features and functions in C++ programming 2. Waste Management System Currently in Malaysia, handling Waste Management System is not fully automatic and cost-effective. Hence, garbage collector may not be able to ensure the maximum cost-effectiveness while performing garbage collection. Below is the description of the scenarios for the garbage collection company that you have to implement the system that you are going to develop using C++ programming in order to solve the problems. - Problem In developing countries such as Malaysia, garbage (i.e. waste and/or rubbish) is being collected mostly when it is overdue or earlier than the time that is necessary. For example, during peak crowded seasons such as during festivals and public holidays, an efficient Automatic Waste Management System is needed to ensure garbage can be collected on time to prevent environmental hazards or health hazards. Consequences of this problem If the above problem could not be solved or mitigated, the probable consequences are elaborated as below: (1) If garbage is not collected on time, it can incur (i) Bacterial growth in the rubbish gathering site(s); (ii) Foul smell; and (iii) Passersby tend to litter their waste/rubbish along the roads, streets and/or at nearby sites or premises since the garbage/rubbish bins are all too full. (2) If garbage is collected too early, it can incur (i) Excessive unnecessary consumption of garbage collector’s resources such as fuel, extra trip(s) for garbage collection, extra wages for the drivers’ wages and etc.; and (ii) The next garbage collection trip may be ill-timed and/or ill-arranged. - Solution A good and efficient Automatic Waste Management System is required to solve the above problems. Solution Description Create a fixed/predefined and/or dynamic distance from the main garbage collection company/enterprise site to each assigned Garbage Location site, in addition to distance from one garbage location’s site to another garbage location’s site. Determine cost, i.e. expenses, incurred in terms of time, fuel consumption, and wages based on the data and information given and assumed by you. You can refer to the suggested data and information below as a proposed example while developing your system. Expenses incurred per trip while garbage collection is being conducted at different waste/garbage locations will be calculated. However, you can treat certain expenses as optional if you do not wish to create too many variables. Nevertheless, delivery distance is an essential variable. The suggested or assumption details are listed as below (this is just a proposed assumption, it is not necessary that you will have to follow the assumed scenarios): (i) Distance needed (as assumed in your scenarios): per km; Fuel consumption (suggested fare): 2.50 RM per km; Drivers’ wages (as assumed in your scenarios if this variable is within the staff payroll or assume RM10.00 per hour for the daily-paid drivers). You can assume that every trip, driver may or may not visit a maximum number of waste locations, no matter whether the distance is near or far from the garbage collection company/enterprise. If Non-Optimized / Regular route is chosen, as long as the garbage/waste level at a garbage gathering location is >=40%, then garbage collection will be performed within 24 hours if the distance from the garbage collector site to the garbage location site is <=30km. (ii) If Optimized route is chosen, only if the garbage/waste level at a garbage gathering location is >=60%, then garbage collection will be performed within 24 hours but only if the distance from the garbage collector site to the garbage location site is <=20km. (iii) If Traveling Salesman Problem (TSP) route is chosen, only if the garbage/waste level at a garbage gathering location is >=40%, then garbage collection will be performed within 24 hours but only if the distance from the waste collector site to the waste location site is <=15km. The Headquarter site of the waste collection is the original departing site, and the driver should come back to the original departing HQ site, i.e. after waste collection, the arrival site should be the same as the departing site. (iv) If Minimum Spanning Tree (MST) route is chosen, only if the garbage/waste level at a garbage gathering location is >=40%, then garbage collection will be performed within 24 hours but only if the distance from the waste collector site to the waste location site is <=15km. Different from TSP route, the arrival site is not necessary to be the same as the departing site. (v) If Greedy route is chosen, regardless of the distance from the garbage collector site to the garbage location site is how far or how near, as long as the garbage/waste level at a garbage gathering location is >=30%, then garbage collection will be performed within 24 hours. - Waste Locations Create a class named as WasteLocations in which 8 or 10 or more locations are saved. Every time when your C++ program is run, each garbage location site should auto generate the garbage/waste level in percentage (through a random number generator and/or the like). (I) Display all relevant information so far, i.e. garbage locations, distances, garbage/waste level (in %) shown, then saved all this information in a file (e.g .txt file, give this file a meaningful name, e.g. WL\_RegularRoute). (II) Display how much cost is being saved by using optimized route or then by MST, TSP or greedy route. So, create another three or more classes named as MSTRoute, TSPRoute, OptimizedRoute and GreedyRoute. This is just an example if you decide to include FIVE types of routes in the system that you are going to develop. (III) User(s) of your system should be able to interact with your displayed information by approving Regular (Non-optimized), Optimized, MST, TSP or Greedy routes (e.g. press ‘Y’ to proceed with the optimized route, otherwise, non-optimized, MST, TSP or greedy route will be followed ‘N’ choice being pressed). - Cost-Effectiveness The waste management efficiency gained by the garbage collector (i.e. the lapse of time for the garbage collection is just right. That is, not too long or too short for the garbage/waste collection since depending on the automatic system that updates the garbage management server.) - Information/data assumed All information and data that is assumed by you, although fictitious, should make sense and logically feasible. All assumptions made by you should clearly show the different approaches of C++ program coding while the user chooses Optimized route, Non-Optimized/Regular route and/or MST and/or TSP and/or Greedy routes in the system you are going to develop. Advice: Please take note that above data are just suggested examples and assumption. Student can increase or reduce the number of variables involved while developing the C++ program. Note: In the above scenarios, there are FIVE TYPES of collection routes (and hence relevant algorithms as mentioned above to be developed) that are suggested to be developed. As mentioned above, you may choose to include TWO or THREE or FOUR TYPES of approaches only for the system that you are going to develop (if you think that FOUR TYPES of approaches are a bit or too complicated and/or a bit too much work for you). After you have decided the number of routes that your system should adopt, it is advised that you should try to further explore the characteristics of Optimized, Non-Optimized, MST and/or TSP and/or Greedy routes so that each driver should know which route to adopt each time after receiving instructions from the Garbage Management Server center at Headquarter of the Garbage Collection company or enterprise. 3. Proper Documentation of your Ideas and Concepts in your report (i) Describe fully if you have to make any assumption(s) while setting the Optimized , Non-optimized , MST, TSP and/or Greedy route approaches to your fictitious garbage collection company or enterprise in the system you are going to develop. (ii) A full description of your proposed opinions, ideas and concepts for your C++ program to the system you are going to develop. (iii) Proper references of the kinds of resources that you have read and from which you have acquired so as to allow you to implement these ideas in order to develop the C++ program. (iv) Describe fully why you think the proposed ideas are important in terms of motivating you to implement them while developing the Garbage Management System so as to provide cost-effectiveness for the fictitious garbage collector company or enterprise.

A functional C++ program involving optimization and/or non-optimization routes, and/or MST and/or TSP and/or greedy routes, primitive/user-defined data types, data structures, other associated structures/data structures, basic and application functions, control statements and etc. (10 marks). 3. Level of Sophistication and Professional Deployment of the developed C++ system, program designs, concepts and ideas (15 marks).

Please do not hold back your innovation and intuitiveness just because you find it difficult to implement (or for any other reason) certain features and/or functions in your C++ program. This is your chance to take a step towards contributing to the society by being innovative and try to be able to deal with real-world daily routine tasks. You might take it further even after your studies. The following suggested areas will be taken into account for each part of the assessment for each coursework as assigned (wherever applicable): ⚫ Innovation, Comprehensiveness and Novelty/Creativity of the developed C++ program to the fictitious case study ⚫ Level of Sophistication and Comprehensiveness of the developed program ⚫ Impact and significance of the daily routine tasks and problem(s) as well as your attempt to the solution(s) and ways of handling and enhancing the daily regular routine tasks. Demonstration of knowledge of the area ⚫ Quality of the concept, including appropriateness and novelty ⚫ Quality of the technological design, including appropriate use of software design concepts, and appropriate good coding practice (abstraction, commenting, naming, and the like) ⚫ Quality of the realization, including how well it works and elaborations over and above the basic requirements ⚫ Including all of the above aspects, clarity of structure, quality of argument / evidence, and insight / novelty.