1. **"Imagine you are leading the development of a software system for a car rental agency. The system needs to manage various types of vehicles in the rental fleet, each with specific attributes and behaviors. Your task is to design and implement a C++ program that leverages inheritance to model these vehicles efficiently.**
2. Start by designing an abstract base class called 'Vehicle.' Define fundamental attributes that are common to all vehicles, such as 'make,' 'model,' and 'year.' Create a virtual function, 'calculateRentalCost,' which should be overridden in derived classes. **(3 marks)**
3. Develop three derived classes: 'Car,' 'SUV,' and 'Truck.' Each derived class should inherit from the 'Vehicle' base class and introduce attributes and behaviors unique to their type. For example, 'Car' might have a 'numDoors' attribute, while 'Truck' may have a 'cargoCapacity' attribute. **(3 marks)**
4. **Imagine you are tasked with developing a C++ program for an online exam system used by a university. The system should support different types of exams, including multiple-choice exams and essay exams, each with specific attributes and behaviors. Additionally, the system should handle exceptions gracefully to ensure reliable exam taking. Your task is to design and implement this program.**
5. Start by designing an abstract base class called 'Exam' that represents common attributes of all exams. Include member variables for 'examID,' 'subject,' and 'duration.' Define a pure virtual function, 'gradeExam,' that should be implemented in derived classes.

**(3 marks)**

1. Create two derived classes, 'MultipleChoiceExam' and 'EssayExam,' both inheriting from the 'Exam' base class. Introduce attributes specific to each type of exam, such as 'questions' for 'MultipleChoiceExam' and 'topic' for 'EssayExam.' **(4 marks)**
2. Implement constructors and destructors for each class to manage resources correctly. Ensure that constructors initialize member variables appropriately, and destructors release any dynamically allocated memory. **(4 marks)**
3. Implement the 'gradeExam' function in both 'MultipleChoiceExam' and 'EssayExam' classes to simulate grading the exams. For 'MultipleChoiceExam,' generate random scores based on the number of correct answers, and for 'EssayExam,' implement a function that allows a grader to assign a score. **(4 marks)**
4. Incorporate exception handling into your program to handle potential errors gracefully. Create custom exception classes (e.g., 'InvalidExamDurationException,' 'GradingErrorException') and handle scenarios such as invalid exam durations or errors during grading. **(4 marks)**
5. In your main program, create instances of both 'MultipleChoiceExam' and 'EssayExam.' Simulate exam-taking and grading scenarios to test your classes and exception handling. Demonstrate how your program handles various exceptions. **(3 marks)**
6. Ensure that your program provides clear and informative output, including exam details, results, and any error messages related to exception handling. **(2 marks)**

Submit your work on or before 29th JUNE 2024.