Consider a company called "Acme Corporation" that employs people from various departments, such as marketing, sales, HR, engineering, and finance. The company has a system that manages the employee database and tracks information such as employee details, job title, salary, performance reviews, and department. To implement this system, we can use object-oriented programming (OOP) concepts such as polymorphism, inheritance, and generics. We can define a base class called "Employee" that has common attributes and methods for all employees, such as name, employee ID, job title, and salary. This class can also have virtual methods for viewing salaries, calculating salary increases, and conducting performance reviews, which can be overridden by derived classes. For example, an employee in the sales department might have a different salary increase formula compared to an employee in the engineering department. We can then define derived classes for each department, such as "MarketingEmployee", "SalesEmployee", "HREmployee", "EngineeringEmployee", and "FinanceEmployee". These classes can inherit from the base class "Employee" and add additional attributes and methods specific to their department. For example, an employee in the marketing department might have a marketing campaign ID, while an employee in the engineering department might have aYearsOfExperience. To store and manipulate the employee data, we can use generics to define a data structure, such as a list, that can store objects of any type derived from the base class "Employee". This allows us to easily add, remove, and modify employees of any department without having to write separate code for each department.

However, in the process of managing the employee database, there may be situations where exceptions occur. For example, if an HR employee tries to access the salary data of an employee in the engineering department, this should not be allowed. In this case, we can define a custom exception class called "AccessDeniedException" that inherits from the built-in "Exception" class. This exception can be thrown whenever an employee tries to access data that they are not authorized to access. Another situation where exceptions may occur is when conducting performance reviews for employees. For example, if an employee has been with the company for less than three years, he may not be eligible for a salary increase. In this case, we can define a custom exception class called "IneligibleForSalaryIncreaseException" that is thrown when an employee is not eligible for a salary increase.

The codes are already given. Incorporate the following modifications.

1. If an exception occurs while running the instructions from the Main function, it will not halt the program. Instead, it will display the error message and perform the subsequent instruction.
2. Add more tests (at least 5).
3. Find the possible bugs and the potential fixes.