1) Test the below concepts by writing necessary classes and methods.
   Single try..catch..finally block
   Multiple catch block
   Nested try catch blocks
   Usage of throw and throws keywords.
2) Create a checked exception invaludAgeException. Override all the constructors from the super class.
3) Create a RuntimeException WrongInputException. Override all the constructors from the super class.
4) Create the below two classes. LearningModule class is dependent on AgeValidator class to check whether the age is valid or not.

LearningModule

> Public String getLearningCode(String Name,age) throws WrongInputException.
>
> **Responsibility:**
> This method accepts Name and age of the person and returns generates a learning code by concatenating name and ageCode. (call getAgeCode from AgeValidator to get age code) In case of InvalidAgeException throw WrongInputException by wrapping InvalidAgeException.

AgeValidator

> Public String getAgeCode(int age) throws InvalidAgeException.
>
> Responsibility:
>
> This method accepts age and returns "Adult" if the age is greater than 18 and returns "kid" if the age is between 0 10 18. If the age is less than 0 then throw InvalidAgeException.

Create the Main class for accepting the data from the user and call the methods on LearningModule=object. In case of any exceptions catch the exceptions and display the appropriate message to the user.

5) In question no.4 modify the interface definition as mentioned below.
                        Storage(Interface)

> Void addEmployee(Employee e) throws EmployeeAlreadyExistsException
> Employee getEmployee(int empno)) throws EmployeeNotFoundException

- While adding the employee if the employee with the same id already exists in the array Raise EmployeeAlreadyExistsException
- While searching if the Employee with the given empno is not found in the array then raise EmployeeNotFoundException.

    Modify other classes accordingly.