Coding Activity: GOAIR Security

In the fast-paced environment of GOAIR, a prominent Indian airline, maintaining security and efficiency at the airport is paramount. The challenge is to ensure that only authorized employees can access restricted areas. To achieve this, GOAIR has turned to you, a skilled software consultant, to automate the process of validating employee entries.

Consider the following scenario:

You receive input that includes the employee's id, entry type, duration in the format of a string in the UserInterface class.

Component Specification: EntryUtility

Type (Class)	Method	Parameters	Responsibilities
EntryUtility	validateEmployeeId	String employeeId	This method validates the employee id according to the condition . If employee id is valid, it returns the true. If employee id is not valid, throw an InvalidEntryException with the appropriate error message. Validation Condition: The employeeId must be exactly ten characters. The first five characters

You receive input that includes the employee's id, entry type, duration in the format of a string in the UserInterface class.

Component Specification: EntryUtility

Type (Class)	Method	Parameters	Responsibilities
EntryUtility	validateEmployeeId	String employeeId	This method validates the employee id according to the condition . If employee id is valid, it returns the true. If employee id is not valid, throw an **InvalidEntryException** with the appropriate error message. **Validation Condition:** The employeeId must be exactly ten characters. The first five characters must be "GOAIR" followed by slash and followed by 4 digits. **Example: GOAIR/7385**
EntryUtility	validateDuration	int duration	This method validates the duration according to the condition. If duration id is valid, it returns the true. If duration is not valid, throw an **InvalidEntryException** with the appropriate error message. **Validation Condition:** The **duration** must between 1 and 5 (both inclusive)

Component Specification: InvalidEntryException (This class inherits the Exception Class)

Type (Class)	Responsibilities
InvalidEntryException	Provided with a single-argument constructor: public **InvalidEntryException** (String message). It is thrown when the employeeId or duration does not follow the validation rules.

Component Specification: InvalidEntryException (This class inherits the Exception Class)

Type (Class)	Responsibilities
InvalidEntryException	Provided with a single-argument constructor: public InvalidEntryException (String message). It is thrown when the employeeId or duration does not follow the validation rules.

Component Specification: UserInterface

Type (Class)	Responsibilities
UserInterface	In the main method get the employee's id, entry type and duration from the user , invoke validateEmployeeId method and validateDuration method and if both are valid display the message, "Valid entry details". If the validation is invalid handle the exception appropriately.

Note:

- Propagate the exceptions that occur in the **EntryUtility** class and handle them in the main method.

- If the employeeId or duration or both employeeId and duration is invalid, display the message "**Invalid entry details**".

- In the sample input and output provided, the highlighted text in bold corresponds to the input given by the user, and the rest of the text represents the output.

exception appropriately.

Note:

- Propagate the exceptions that occur in the **EntryUtility** class and handle them in the main method.

- If the employeeId or duration or both employeeId and duration is invalid, display the message "**Invalid entry details**".

- In the sample input and output provided, the highlighted text in bold corresponds to the input given by the user, and the rest of the text represents the output.

- Ensure to follow the object-oriented specifications provided in the question description.

- Ensure to provide the names for classes, attributes, and methods as specified in the question description.

- Adhere to the code template, if provided.

- Please don't use System.exit(0) to terminate the program.

Sample Input / Output 1:

Enter the number of entries

3

Enter entry 1 details

- **Please don't use System.exit(0) to terminate the program.**

Sample Input / Output 1:

Enter the number of entries

3

Enter entry 1 details

GOAIR/8924:Security:4

Valid entry details

Enter entry 2 details

GOAIR/1353:Inpection:2

Valid entry details

Enter entry 3 details

GOAIR/1353:CheckIn:5

Valid entry details

Sample Input / Output 2:

Enter entry 3 details

GOAIR/1353:CheckIn:5

Valid entry details

Sample Input / Output 2:

Enter the number of entries

2

Enter entry 1 details

AIRIND/8924:Boarding:4

Invalid entry details

Enter entry 2 details

GOAIR/1353:Inpection:7

Invalid entry details

Coding Activity: Shop Validator

Description   Submission   Code Editor   Grading view

TechStore is a gadget shop that sells various electronic gadgets such as laptops, smartphones, tablets, and accessories. The store aims to provide consumers with the latest technology products, with product warranties ranging from 6 months to 36 months.

As a software consultant, you have been asked to automate the process of stock updates for all products within the specified warranty range.

The stockkeeper will enter the product information as a String with the format *gadgetID:gadgetType:warrantyPeriod.*

Component Specification: GadgetValidatorUtil Class

Type (Method)	Responsibilities
public boolean validateGadgetID(String gadgetID)	This method takes gadgetId as an argument and validates it. The gadgetId must start with an uppercase letter followed by a 3-digit number.If valid, return true else throw *InvalidGadgetException* with the message "*Invalid gadget ID*". Example: J367.
public boolean validateWarrantyPeriod(int	This method validates the *warrantyPeriod* argument and updates the stock for products with warranty periods between 6 and 36 months. If the warranty period is outside this range, it raises an

Component Specification: GadgetValidatorUtil Class

Type (Method)	Responsibilities
public boolean validateGadgetID(String gadgetID)	This method takes gadgetId as an argument and validates it. The gadgetId must start with an uppercase letter followed by a 3-digit number. If valid, return true else throw *InvalidGadgetException* with the message "*Invalid gadget ID*". Example: J367.
public boolean validateWarrantyPeriod(int period)	This method validates the *warrantyPeriod* argument and updates the stock for products with warranty periods between 6 and 36 months. If the warranty period is outside this range, it raises an *InvalidGadgetException* with the message "*Invalid warranty period*".

Component Specification: InvalidGadgetException Class (inherits Exception class)

Type (Class)	Responsibilities
InvalidGadgetException	Provided with a single-argument constructor: public InvalidGadgetException(String message). Thrown when the gadgetId and warrantyPeriod;does not follow the below validation rules.

In the UserInterface class, write the **main** method to get input from the user, as shown in the sample input. Split the input string using the delimiter (:) to extract the gadget details.

Invoke the **validateGadgetID** and **validateWarrantyPeriod** methods, passing *gadgetId* and *warrantyPeriod* as parameters, respectively. Based on validation results (either true for successful validation or an exception for validation errors), you should display specific messages:

In the UserInterface class, write the main method to get input from the user, as shown in the sample input. Split the input string using the delimiter (:) to extract the gadget details.

Invoke the validateGadgetID and validateWarrantyPeriod methods, passing *gadgetId* and *warrantyPeriod* as parameters, respectively. Based on the validation results (either true for successful validation or an exception for validation errors), you should display specific messages:

- If validation succeeds without exceptions, print "*Warranty accepted, stock updated*".
- If an exception is raised during validation, print the appropriate error message related to the validation issue.

Use the exception handling mechanism in the main method to handle the exception. Use catch blocks for handling the exception. In the catch block, display the appropriate messages.

Note:
- **Try, and catch blocks are mandatory.**
- In the sample input and output provided, the highlighted text in bold corresponds to the input given by the user, and the rest of the text represents the output.
- The class and methods should be declared as public, and all the attributes should be declared as private.
- Ensure to follow the object-oriented specifications provided in the question description.
- Ensure to provide the names for classes, attributes, and methods as specified in the question description.
- Adhere to the code template, if provided.
- Please do not use System.exit(0); to terminate the program.

Sample Input and Output 1

Enter the number of gadget entries

Sample Input and Output 1

Enter the number of gadget entries

2

Enter gadget 1 details

T283:Television:10

Warranty accepted, stock updated

Enter gadget 2 details

L123:Laptop:24

Warranty accepted, stock updated

Sample Input and Output 2

Enter the number of gadget entries

2

Enter gadget 1 details

S4567:Smartphone:40

Invalid gadget ID

Warranty accepted, stock updated

Enter gadget 2 details

L123:Laptop:24

Warranty accepted, stock updated

Sample Input and Output 2

Enter the number of gadget entries

2

Enter gadget 1 details

S4567:Smartphone:40

Invalid gadget ID

Enter gadget 2 details

P345:Printer:40

Invalid warranty period