**Scala Assignment**

* **Problem Statement 1:**

Generate solution for you are tasked with creating a random password generator in Scala. The generator will take user input for password length and generate a random password that includes a mix of lowercase letters, uppercase letters, numbers, and special characters.

* **Problem Statement 2: UST Shopping Cart Application**

You are tasked with developing a Shopping Cart application in Scala. The application will manage a shopping cart, allowing customers to add, remove, update, view items in their cart, and proceed to payment. Each item will have details such as name, quantity, price, and category. Additionally, users will be able to make payments through a simulated payment gateway – Credit Car, Debit Card, UPI. The application will also calculate the total price including GST (Goods and Services Tax) and will add delivery charges below than Rs.200 cart value.

**Item Class:**

Create an **Item () class** with the following attributes:

* id: Unique identifier for the item
* name: Name of the item.
* quantity: Quantity of the item.
* price: Price of the item.
* category: Category of the item.

**Shopping Cart:**

Create a **ShoppingCart()** class that manages a collection of Item objects. Implement the following methods:

* addItem(item: Item): Adds a new item to the cart.
* updateItem(id: Int, updatedItem: Item): Updates an existing item in the cart.
* removeItem(id: Int): Removes an item from the cart.
* viewCart(): Displays all items in the cart.
* totalPrice(withGST: Boolean = true): Calculates and displays the total price of all items in the cart with GST charges.

**Payment Gateway:**

* Create a PaymentGateway class to simulate payment processing. Implement the following methods:  
  **processPayment(amount: Double, paymentMethod: String):** Simulates processing a payment and returns a confirmation message.
* Payment methods can include "Credit Card", "Debit Card", and "UPI".

**GST Calculation:**

* Assume a GST rate of 5%. Implement the GST calculation within the totalPrice method.

**Cart Operations:**

Implement the following functions:

* Add Item: Allow users to input details for a new item and add it to the cart.
* Update Item: Allow users to update the details of an existing item.
* Remove Item: Allow users to remove an item from the cart.
* View Cart: Display all items in the cart.
* Calculate Total: Calculate and display the total price of items in the cart, optionally including GST.
* Make Payment: Allow users to proceed to payment and select a payment method. Use the PaymentGateway class to process the payment.

**Error Handling:**

Ensure appropriate error handling for scenarios such as trying to update or remove an item that doesn't exist, and handling payment errors.

Sample Outcome:

Welcome to the UST Shopping Cart!

Please choose an option:

1. Add a new item

2. Update an existing item

3. Remove an item

4. View cart

5. Calculate total price

6. Make payment

7. Exit

Option: 1

Enter item name: Apple

Enter quantity: 3

Enter price: 50.00

Enter category: Fruits & Vegetables

Item added successfully!

Option: 4

Viewing cart:

1. [1] Apple - Quantity: 3, Price: 50.00, Category: Fruits & Vegetables

...

Option: 5

Cart Value: ₹150.00

Deliver Charge: ₹30.00

GST: ₹7.50

**Amount Payable: ₹187.50**

Option: 6

Choose a payment method (Credit Card/Debit Card/UPI): Credit Card

Processing payment of 187.50 using Credit Card...

Payment successful! Confirmation number: UST123456789

Option: 7

Exiting the application. Goodbye!

* **Problem Statement 3: Case Classes and Pattern Matching**

Create a Scala application that uses case classes to model a simple payroll system. Implement pattern matching to calculate the salary of different types of employee – FullTimeEmployee, PartTimeEmployee, ContractType, Freelancers.

* **Problem Statement 4: File Processing**

Write a Scala program to read a text file, count the occurrences of each word, and display the top N most frequent words.

* Create a method wordCount(filePath: String, topN: Int): List[(String, Int)] that reads a text file and returns a list of tuples containing the top N most frequent words and their counts.
* Program ask user to enter N top most frequent words and show N most frequent words as output.
* **Problem Statement 5: File Analysis Application in Scala**

The application will process a text file and provide various analytical insights about its content. The insights will include word count, line count, character count, frequency of each word, and the top N most frequent words.

* FileAnalyzer Class: Create a FileAnalyzer class with the following methods:
* loadFile(filePath: String): Load and Read a text file.
* wordCount(): Returns the total number of words in the file.
* lineCount(): Returns the total number of lines in the file.
* characterCount(): Returns the total number of characters in the file.
* averageWordLength(): Double: Returns the average word length in the file.
* mostCommonStartingLetter(): Option[Char]: Returns the most common starting alphabet of words in the input files.
* wordOccurrences(word: String): Int: Returns the number of occurrences of a specific word in file.