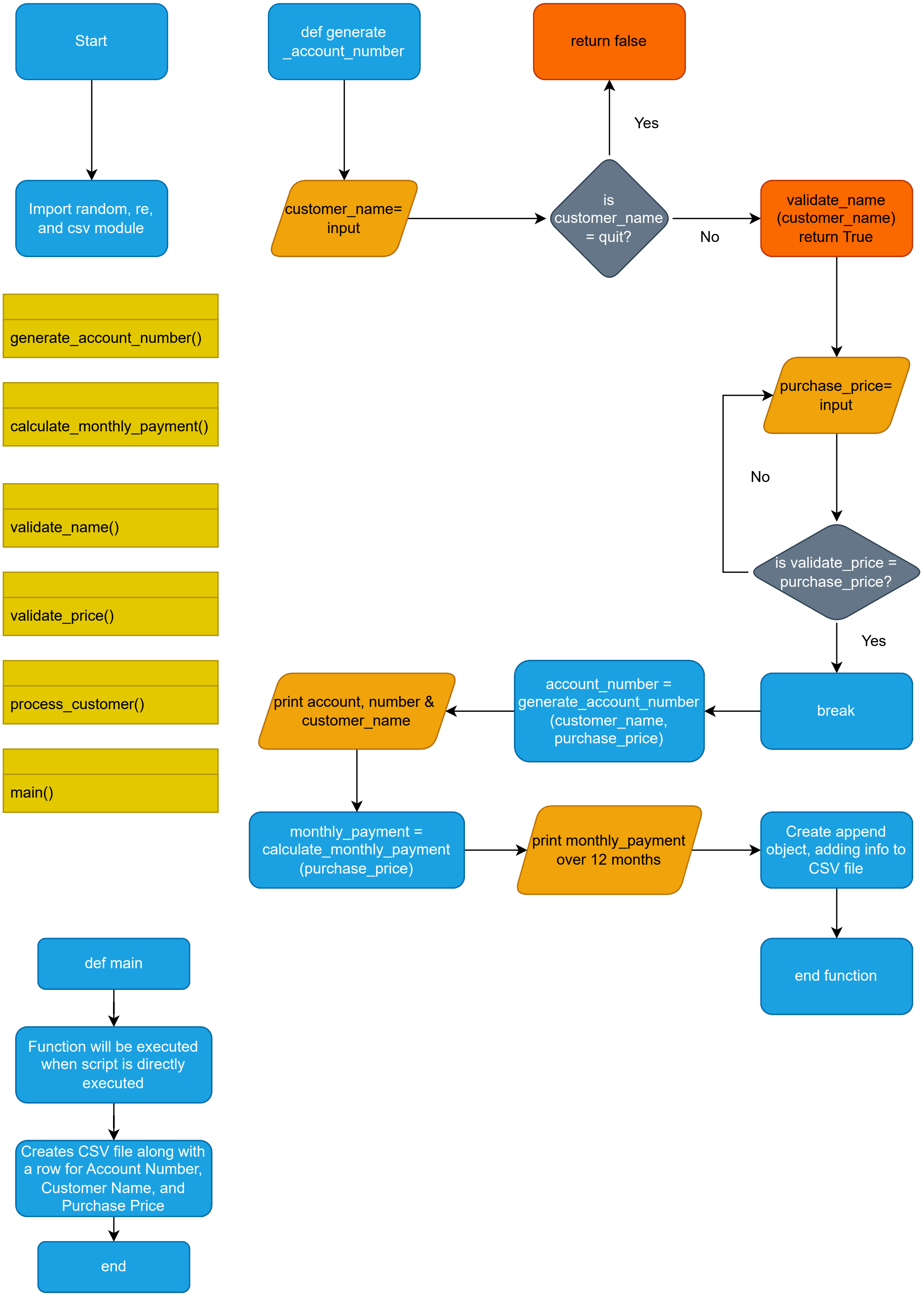
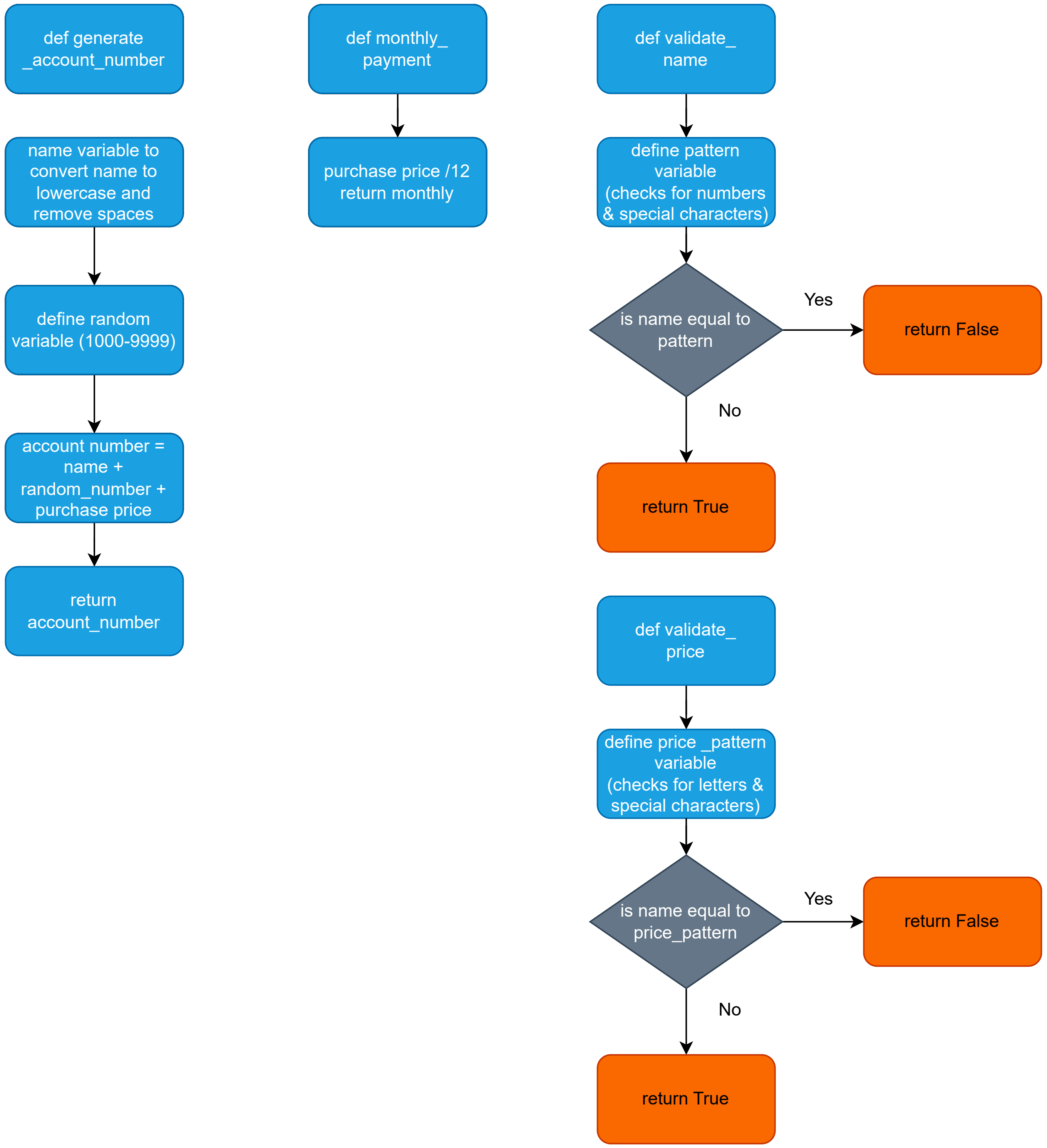

Start
Import random, re, and csv module
generate_account_number()
calculate_monthly_payment()
validate_name()
validate_price()
process_customer()
main()
def generate _account_number
customer_name= input
is customer_name = quit?
Yes
return false
No
validate_name (customer_name) return True
purchase_price= input
No
is validate_price = purchase_price?
Yes
break
account_number = generate_account_number (customer_name, purchase_price)
print account, number & customer_name
monthly_payment = calculate_monthly_payment (purchase_price)
print monthly_payment over 12 months
Create append object, adding info to CSV file
end function
def main
Function will be executed when script is directly executed
Creates CSV file along with a row for Account Number, Customer Name, and Purchase Price
end


def generate _account_number
name variable to convert name to lowercase and remove spaces
define random variable (1000-9999)
account number = name + random_number + purchase price
return account_number
def monthly_ payment
purchase price /12 return monthly
def validate_ name
define pattern variable (checks for numbers & special characters)
is name equal to pattern
Yes
return False
No
return True
def validate_ price
define price _pattern variable (checks for letters & special characters)
is name equal to price_pattern
Yes
return False
No
return True

```
# Import necessary modules
Import random
Import re
Import csv

# Function to generate an account number
Function generate_account_number(name, purchase_price):
    # Convert name to lowercase and remove spaces
    name = Lowercase(name)
    name = RemoveSpaces(name)

    # Generate a random 4-digit number
    random_number = RandomNumberBetween(1000, 9999)

    # Create the account number using name, random number, and purchase price
    account_number = GetFirstThreeCharacters(name) + ConvertToString(random_number) + ConvertToString(Integer(purchase_price))

    Return account_number

# Function to calculate monthly payment
Function calculate_monthly_payment(purchase_price):
    Return purchase_price / 12

# Function to validate customer name
Function validate_name(name):
    pattern = "^[a-zA-Z ]+$"
    If Not Match(pattern, name):
        Print "Invalid name. Please enter only alphabetical characters and spaces."
        Return False

    Return True

# Function to validate purchase price
Function validate_price(price):
    price_pattern = "^\D+$"
    If Not Match(price_pattern, price):
        Print "Invalid price. Please enter a numeric value."
        Return False

    Return True

# Function to process customer information
Function process_customer():
    # Prompt for customer name
    customer_name = Input("Enter customer name (or enter 'quit' to exit): ")

    # Check if customer wants to quit
    If Lowercase(customer_name) == "quit":
        Return False

    # Validate customer name
    If Not validate_name(customer_name):
        Return True

    # Loop until a valid purchase price is entered
    While True:
        purchase_price = Input("Enter purchase price: ")

        # Validate purchase price
        If validate_price(purchase_price):
            Break

    # Convert purchase price to float
    purchase_price = ConvertToFloat(purchase_price)

    # Generate account number
    account_number = generate_account_number(customer_name, purchase_price)

    # Print customer information
    Print "Account Number:", account_number
    Print "Customer Name:", customer_name
    # Calculate and print monthly payments
    monthly_payment = calculate_monthly_payment(purchase_price)
    Print "Monthly Payments for the next 12 months:"
    For month in Range(1, 13):
        Print "Month {}: ${:.2f}".Format(month, monthly_payment)

    # Write customer information to CSV file
    Open "customer_data.csv" in append mode
    Create a CSV writer object
    Write a new row with account number, customer name, and purchase price

    Return True

# Main function
Function main():
    # Open the CSV file and write the header row
    Open "customer_data.csv" in write mode
    Create a CSV writer object
    Write the header row ["Account Number", "Customer Name", "Purchase Price"]

    # Loop to process customers
    While True:
        # Process customer information
        If Not process_customer():
            Break

# Execute the main function if this script is run directly
If ThisScriptIsRunDirectly():
    Run the main function
```