Computer Programming

Final Project (**Vending Machine**)

Team (21)

	Name	ID	Sec
1	Abdalla Nagi Abdalla Osman (L)	2002356	3
2	Ahmed Anas Mohamed Ahmed	2101082	1
3	Mohamed Hassan Fathallah Hussein	1807176	3
4	Mohamed Osama Mahmoud	1901501	3

Commented Code analysis:

1. // Struct to hold date information

   Struct to split the date into days, months, and years.

```
// Struct to hold date information
struct Date {
    int day;
    int month;
    int year;
};
```

2. // Struct to represent a product

   Struct to show what are the information contained in the product.

```
// Struct to represent a product
struct Product {
    string name;
    double price;
    int count;
    Date expirationDate;
    string outlet;
};
```

3. // Class for vending machine operations

   Defining a class for the vending machine that includes:

   - **Private part** to which the user has no access to it:
   - the vector named "products" which has all the products in the vending machine.

```
vector<Product> products;
```

   - Function to get the current date using <ctime> library.

```
// Function to get the current date
Date getCurrentDate() {
    time_t now = time(0);  // Get current system time
    tm *currentTime = localtime(&now); // Convert system time to local time

    // Extract day, month, and year from local time
    Date currentDate;
    currentDate.day = currentTime->tm_mday;
    currentDate.month = currentTime->tm_mon + 1;
    currentDate.year = currentTime->tm_year + 1900;

    return currentDate;
}
```

❖ **Public part** to which the user has access to it.

- Function to add a new product to the vending machine.

```
// Function to add a new product to the vending machine
void addProduct() {
    Product newProduct;
    cout << "Enter product name: ";
    cin >> newProduct.name;
    cout << "Enter product price: ";
    cin >> newProduct.price;
    cout << "Enter product count: ";
    cin >> newProduct.count;
    cout << "Enter day of expiration : ";
    cin >> newProduct.expirationDate.day;
    cout << "Enter month of expiration : ";
    cin >> newProduct.expirationDate.month;
    cout << "Enter year of expiration : ";
    cin >> newProduct.expirationDate.year;
    cout << "Enter outlet: ";
    cin >> newProduct.outlet;

    products.push_back(newProduct);
    cout << "Product added successfully!\n";
}
```

- Function to display the available products in the vending machine.

```
// Function to display available products
void displayAvailableProducts() {
    cout << "Available Products:\n";
    for (const auto &product : products) {
        if (product.count > 0 && isAvailable(product)) {
            cout << product.name << ", Price: " << product.price << endl;
        }
    }
}
```

- Function to check if a product is available based on expiration date and count.

```
// Check if the product has expired or there are no available items
if (expirationDate.year < currentDate.year ||
    Date VendingMachine::isAvailable::expirationDate ationDate.month < currentDate.month) ||
    (expirationDate.year == currentDate.year && expirationDate.month == currentDate.month
     && expirationDate.day < currentDate.day)) {
    return false; // The product has expired
}

return product.count > 0; // Check if there are available items
}
```

- Function to select a product.

```
// Function to select a product
void selectProduct() {
    displayAvailableProducts();
    string selectedProduct;
    cout << "Enter the name of the product you want: ";
    cin >> selectedProduct;

    // Iterate through products to find the selected product
    for (auto &product : products) {
        if (product.name == selectedProduct && product.count > 0 && isAvailable(product)) {
            double depositedMoney;
            cout << "Please deposit " << product.price << ": ";
            cin >> depositedMoney;

            if (depositedMoney >= product.price) {
                // Dispense product and handle change
                dispenseProduct(product);
                if (depositedMoney > product.price) {
                    double change = depositedMoney - product.price;
                    cout << "Change returned: $" << change << endl;
                }
                product.count--;
                return;
            } else {
                cout << "Insufficient funds, Please deposit the correct amount\n";
                return;
            }
        }
    }
    cout << "Invalid selection or product not available.\n";
}
```

- Function to simulate dispensing a product.

```
    // Check if the product has expired or there are no available items
    if (expirationDate.year < currentDate.year ||
        Date VendingMachine::isAvailable::expirationDate ationDate.month < currentDate.month) ||
        (expirationDate.year == currentDate.year && expirationDate.month == currentDate.month
         && expirationDate.day < currentDate.day)) {
        return false; // The product has expired
    }

    return product.count > 0; // Check if there are available items
}
```

4. // Main function to interact with the vending machine

The int main () function that will apply the functions we created in the vending machine class to achieve the desired mission and it includes:

- Loop for vending machine operation modes that keeps repeating and gives the user the ability to change between modes until he chooses to exit the program.

```
// Main function to interact with the vending machine
int main() {
    VendingMachine vendingMachine;
    int mode;

    // Loop for vending machine operation modes
    do {
        cout << "Select Mode: \n";
        cout << "1. Programming Mode\n";
        cout << "2. Operation Mode\n";
        cout << "0. Exit\n";
        cout << "Enter mode: ";
        cin >> mode;

        // Switch between different modes of operation
        switch (mode) {
            case 1:
                vendingMachine.addProduct();
                break;
            case 2:
                vendingMachine.selectProduct();
                break;
            case 0:
                cout << "Exiting...\n";
                break;
            default:
                cout << "Invalid choice!\n";
                break;
        }
    } while (mode != 0);

    return 0;
}
```

Screenshots for all modes and test cases:

```
Select Mode:
1. Programming Mode
2. Operation Mode
0. Exit
Enter mode: 0
Exiting...

Process returned 0 (0x0)   execution time : 111.616 s
Press any key to continue.
```

```
Select Mode:
1. Programming Mode
2. Operation Mode
0. Exit
Enter mode: 1
Enter product name: Cola
Enter product price: 15
Enter product count: 20
Enter day of expiration : 1
Enter month of expiration : 1
Enter year of expiration : 2024
Enter outlet: 2
Product added successfully!
Select Mode:
1. Programming Mode
2. Operation Mode
0. Exit
```

```
Enter mode: 2
Available Products:
Cola, Price: 15
Water, Price: 5
Tea, Price: 7.5
Enter the name of the product you want: Tea
Please deposit 7.5: 8
Dispensing from outlet: 3 - Tea
Change returned: $0.5
Select Mode:
1. Programming Mode
2. Operation Mode
0. Exit
Enter mode: 0
Exiting...

Process returned 0 (0x0)   execution time : 98.817 s
Press any key to continue.
```

- Showing that the vending machine doesn't give the user the ability to pick a product that is out of stock or expired.

```
Select Mode:
1. Programming Mode
2. Operation Mode
0. Exit
Enter mode: 1
Enter product name: Pepsi
Enter product price: 17.5
Enter product count: 20
Enter day of expiration : 20
Enter month of expiration : 12
Enter year of expiration  2023
Enter outlet: 1
Product added successfully!
Select Mode:
1. Programming Mode
2. Operation Mode
0. Exit
Enter mode: 2
Available Products:
Cola, Price: 15
Enter the name of the product you want:
```

```
Select Mode:
1. Programming Mode
2. Operation Mode
0. Exit
Enter mode: 1
Enter product name: cake
Enter product price: 12
Enter product count: 0
Enter day of expiration : 12
Enter month of expiration : 3
Enter year of expiration : 2024
Enter outlet: 6
Product added successfully!
Select Mode:
1. Programming Mode
2. Operation Mode
0. Exit
Enter mode: 2
Available Products:
Enter the name of the product you want: cake
Invalid selection or product not available.
```