Week 4: Constructor, Destructor, Default Parameters, const and static Members, Pass object and return object

Learning Materials: Chapter 6

Task 1

Design a **Calculator** class where the object of the class has a field (or variable) that keeps the **current status of the calculator** (integer type). The class should ensure encapsulation, with all operations being performed on the internal field. In case of invalid arithmetic operations, the object will keep the previous status and display a message "**Invalid arithmetic operation.**".

List of methods for the Calculator class:

Constructor methods:

Zero-argumented constructor initializes the object with 0 value to the current status. One-argument constructor initializes current status with the passed argument.

Get methods:

Write appropriate get and set methods for the **current** status. But make them **private**.

Arithmetic operations:

void add(int value); - Adds the passed value to the current value.
void subtract(int value); - Subtracts the passed value from the current value.
void multiply(int value); - Multiplies the current value by the passed value.
int divideBy(int value); - Divides the current value by the passed value. Return the remainder of the divide operation.

Utility methods:

void clear(): Resets the current value to 0.
void display(): Displays the current value. e.g. "Calculator display: <current status>"

Destructor:

~Calculator(): Displays the current value and message that **"Destructor of the Calculator object is called."**

Sample Output for each line of code:

```
--------------------------------------------------
add(10); display();
Calculator display: 10
--------------------------------------------------
add(7); display();
Calculator display: 17
--------------------------------------------------
multiply(31); display();
Calculator display: 527
--------------------------------------------------
subtract(42); display();
Calculator display: 485
--------------------------------------------------
divide (7); display();
Calculator display: 69
--------------------------------------------------
divide (0); display();
Invalid arithmetic operation.
Calculator display: 69
--------------------------------------------------
add(3); display();
Calculator display: 72
--------------------------------------------------
clear(); display();
Calculator display: 0
--------------------------------------------------
0
Destructor of the Calculator object is called.
```

Task 2

Define a class in C++ with the following description:

Private Members:

- A data member Flight number of type integer
- A data member Destination of type string/ char array
- A data member Distance of type float
- A data member MaxFuelCapacity of type float
- A member function double CalFuel() to **return** the value of fuel required for the current distance value as per the following criteria

Distance	**Fuel**
<=1000	500
more than 1000 and <=2000	1100
more than 2000	2200

Public Members:

A function **FeedInfo()** to allow the user to enter values for Flight Number, Destination, Distance, Max Fuel capacity using a keyboard.
A function **ShowInfo**() to allow the user to view the content of all the data members. At the end, it will also show appropriate messages like "Fuel capacity is fit for this flight distance" or "Not sufficient Fuel Capacity for this flight."

Task 3

Define a class **"BankAccount"** with the following description.
Each account will have the following information:

Private members

- The account number.
- The account holder name.
- The account type (current/savings) (assume the data type)
- The current balance.

- The minimum balance (An account has to maintain the minimum amount; it cannot withdraw. It can only be initialized at the time of creating object. **USE const modifier**)

The class will have the following criteria:.

Public members

- The member variable value of the object can be assigned **during** object **creation** or **after** the object has been created.
- A function to show all the information of a **BankAcccout** object.
- Function **showBalance() (**for displaying current balance**),**
- Functions **deposit()** and **withdrawal()** of money from an account. Show appropriate messages for invalid amount.
- Function **giveInterst()** will deposit net interest to the account. Default interest is **3 percent** of current balance but it might be different. A fixed **10%** Source Tax will be deducted from the incurred interest.
- When the **BankAccount** object is destroyed display a message like : *Account of Mr. X with account no 1234 is destroyed with a balance BDT 5000*

Non-member functions

Define a non-member function **void display_stat();**(this is not a member function of the BankAccount class) that displays the following information:
Total number of BankAccount objects created and total number of BankAccount objects currently present in the program. It also shows the total amount of source tax collected from all the bank accounts.

Define a non-members function with the prototype
BankAccount Larger(**const** BankAccount A, **const** BankAccount B): it returns the BankAccount Object that has a higher current balance.