

Week 3:Defining simple classes

Learning Materials: Chapter 6

Topic:

1. Class Definition.
2. Access Specifier.
3. Accessing Members of an object.

[RULEs]: Member variable is always private. Write member functions and make it public if necessary. Add necessary parameters and return necessary values for each function according to its descriptions. Do not change the name of the function.

Task 1

Create a class named "**Counter**". An object of Counter class keeps track of count. The object also stores the value of increment steps. For example if the increment step is 5, the **count** will be incremented by **5** each time increment is done. Implement the following member functions (task of the function is written after a hyphen):

- **setIncrementStep** - it sets the value of Increment Step in the appropriate member variable. **Do not take input from the user. Use appropriate parameter(s).**
- **getCount** - it returns the current count value.
- **increment** - it increments the count by increment step for that. For example : if the current count is 4 and increment step value is 2 then executing the function will update the count to 6.
- **resetCount** - it resets the value of count to 0.

Task 2

In mathematics, a rational number is a number such as $-3/7$ that can be expressed as the quotient or fraction **p/q** of two integers, a numerator p and a non-zero denominator q.

Create a class **RationalNumber** which stores two data - **numerator** and **denominator** of data type **int**. The object of this class will not store any undefined value. A member function should not change the member

variables to undefined state. Implement the following member functions (task of the function is written after a hyphen):

- **void assign(int numerator,int denominator)** - it sets the value to the data member. **Do not store if it is mathematical undefined and display error message “You can not assign 0 as denominator”.**
- **convert** - it returns the **decimal equivalent** example: $3/2 = 1.5$ so if numerator and denominator are 3 and 2 respectively then return 1.5.
- **invert** - example if the RationalNumber object stores $3/2$ after calling this function the same object will store $2/3$. **Do not invert if the result becomes mathematical undefined and display an error message “You can not assign 0 as denominator. Inversion Failed”.**
- **void print()** - this member function will display the **RationalNumber** object. Example : if the numerator and denominator is 3 and 2 respectively then print() will display in the console -
The Rational Number is 3/2.

Task 3

Create a **Medicine** class. An object of this class should be able to store trade name, generic name, unit price. The unit price which is the maximum retail price. However there can be a discount of a medicine object at any time. The discount can be from 0-30% of the maximum retail price. *[use char array of size 30 to store the trade name, generic name]*

Implement the following member functions (task of the function is written after a hyphen). Assign the default value if the argument is invalid for any member function.

- **void assignName(char name[], char genericName[])** - this member function will initialize the name and genericName data members.
- **void assignPrice(double price)** - this member function will initialize the unit price. *Price needs to be non-negative. Default unit price is 0.*

- **void setDiscountPercent(double percent)** - this member function will initialize the discountPercent. *discountPercent needs to be within 0-30%. Default discount is 5 percent.*
- **double getSellingPrice(int nos)** - this member function returns the selling price of the medicine for given nos of unit price . Selling price = price - discount. If nos is not passed as argument return selling price of single unit.
- **void display()** - this member function displays the information of a medicine object in the console. Example:
**MRP. of Napa (Paracetamol) is BDT 1.00. Current discount 10%.
Selling price BDT 0.90.**