

## 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 them 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 "**Temperature**". An object of the Temperature class keeps track of temperature. The object also stores the value of increment steps. For example if the increment step is 5, the **temperature** 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).
- **getTemperature** - it returns the current temperature value.
- **increment** - it increments the temperature by increment step for that. For example : if the current temperature is 4 and increment step value is 2 then executing the function will update the temperature to 6.
- **resetTemperature** - it resets the value of temperature 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 class to represent a **bank account** which has private data members-**name of the customer, account number, type of account, balance amount in the account.** A customer will be the object of this class. Implement the following member functions (task of the function is written after a hyphen):

- **void customerDetails()**-this member function will initialize the name of the customer and account number.
- **void accountType()** - this member function will initialize the type of the account(Current /saving).
- **void balance()**- this member function will show the balance amount of the customer's account.
- **void deposit()**- this member function will deposit an amount and display the final balance.
- **void withdraw()** - this member function will withdraw an amount after checking the balance.
- **void display()** - this member function displays the information of a customer object in the console.