Week 5: String Class, Array of Objects, Reference Variable, Use of Project feature in CodeBlocks IDE.

Learning Materials: Chapter 7

Demonstration:

1. Create a Project in codeblocks IDE to Implement.
2. Use of reference variables in return type.
3. Array of objects
4. Class String

Task 1:

Create a **SavingsAccount** class. All Saving Account of **annualInterestRate**. Each member of the class contains a private data member **savingsBalance** indicating the **amount the saver currently has on deposit**. Provide a member function **calculateMonthlyInterest()** that calculates the monthly interest by multiplying the balance by **annualInterestRate** divided by 12; this interest should be added to **savingsBalance**.

Provide a **static member function** modifyInterestRate that sets the static annualInterestRate to a new value.

Write a driver program to test the class SavingsAccount. Instantiate two different objects of class SavingsAccount: saver1 and saver2, with balances of $2000.00 and $3000.00, respectively. Set the **annualInterestRate** to **3 percent**. Then calculate the **monthly interest** and print the new balances for each of the savers. Then set the **annualInterestRate** to **4 percent**, calculate the next month's interest and print the new balances for each of the savers. **Also, count the number of objects created and destroyed for a class using static data members and static member functions**

Task 2:

Let the student information system of a school keep the records of every student’s name, date of birth, id and their respective cgpa. Create a class called student that will allow you to store all this information regarding a student.

You should define the **setInfo**() function which will set the necessary information of a student object. (write individual setter function for each member variable and call them from setInfo())
Define a function named **getInfo**() which will return all the stored information belonging to a student object. (write individual getter function for each member variable and call them from getInfo() to display on the console)

Include a constructor function to initialize the student object by zero and null values.
All the member variables to the student class should be private.

Task 3:

Update the task 2 solution to change necessary function as constant member function. Next, include one more **const** member function named **getAge**() which will return the age of a student calculated from date of birth. To accomplish this task you have to take additional user input to take current date.
Your program should check the validity (whether the date is valid or not) of date of birth and current date.

Task 4:

Let employee information system of an institution keeps the records of every employee's name, date of birth and their respective salaries. Create a class called employee that will allow you to store all these information regarding an employee.
Write getter and setter functions for all the member variables. (getter function return the value and setter function assign the value from the parameter to the member variables). Before setting any value to the member variables you need to check for these
Name: The length has to be more than two. Otherwise assign the default name John Doe
Date of Birth: Every employee has an age higher than 18. If an invalid value is given assign 1 January 2002.
Salaries: The salary has to be in between BDT 10000 to BDT 100000. If an invalid value is given assign BDT 10000.

Define setInfo() function which will call all the setter functions to set the necessary information of an employee object.
Define a function named getInfo() which will display all the stored information belonging an employee object using the return value of the getter function.