**Digital Assignment – 2**

**Object Oriented Programming**

**Name: Hari Krishna Shah**

**VIT ID: 21BCS0167**

Ques 1.

1. Implement a C++ program to reverse the case of each alphabet in the given string by overloading the operator ! .

Answer:

#include<iostream>

#include<string.h>

**using** **namespace** std**;**

**class** string\_class

**{**

**char** string**[**100**];**

**public:**

**void** **operator!();** *//Overloaded '!' Operator*

**void** get\_details**()**

**{**

cout**<<"\nEnter the string: ";**

cin**>>**string**;**

**}**

**void** display\_string**()**

**{**

cout**<<**string**;**

**}**

**};**

**void** string\_class**::operator!()**

**{**

cout**<<"\n\n The reversed string is given below: "<<**endl**;**

**int** i **=** 0**;**

**while(**string**[**i**]** **!=** '\0'**)**

**{**

**if(**string**[**i**]>=**65**&&**string**[**i**]<=**96**)**

**{**

cout**<<char(**string**[**i**]+**32**);**

**}**

**else** **if(**string**[**i**]>=**97**&&**string**[**i**]<=**122**)**

**{**

cout**<<char(**string**[**i**]-**32**);**

**}**

i**++;**

**}**

**}**

**int** main**()**

**{**

**class** string\_class str**;**

str**.**get\_details**();**

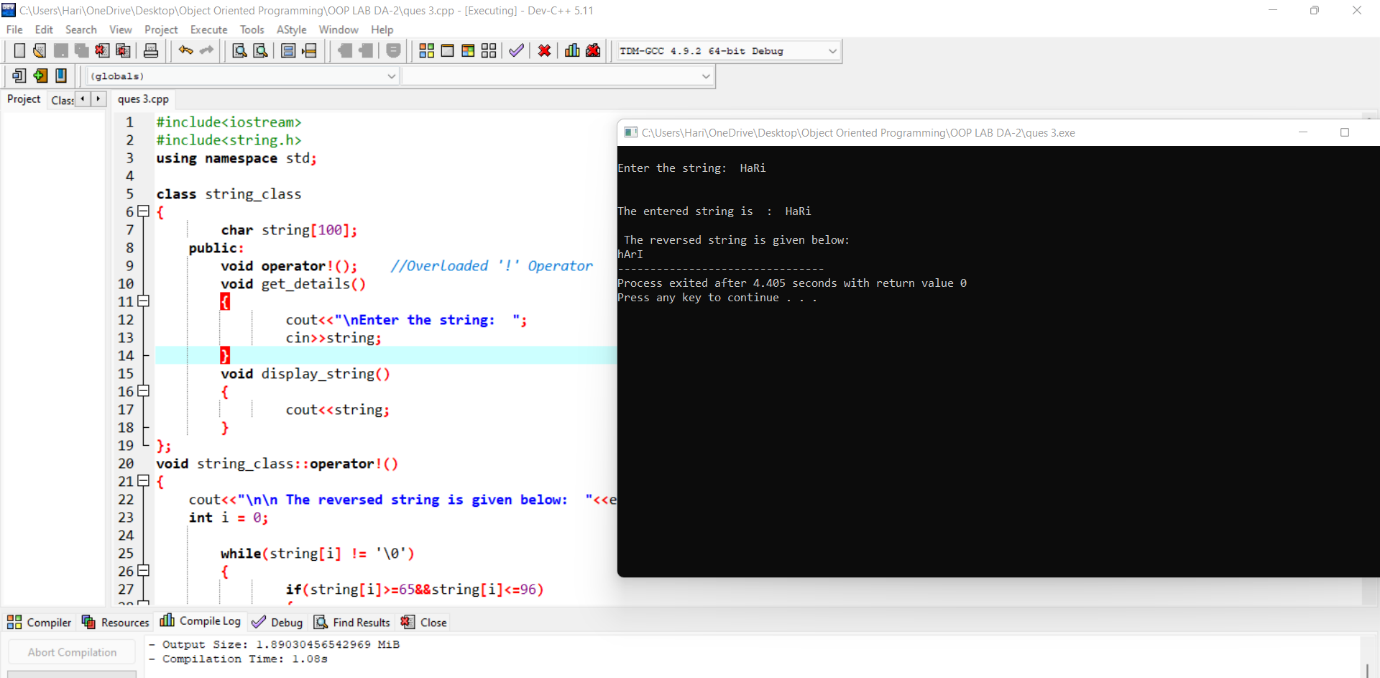
cout**<<"\n\nThe entered string is : ";**

str**.**display\_string**();**

**!**str**;**

**return** 0**;**

**}**



Ques 2.

1. Develop an OOP to perform the assignment = operator overloading to assign one vector into another vector. Define a constructor to allocate the memory space for the vector using dynamic memory allocation. Note: Here, vector represents the single dimensional array which contains the set of values.

Answer:

#include <iostream>

#include <malloc.h>

**using** **namespace** std**;**

**class** vector**{**

**private:**

**int** **\***array**;**

**int** size**;**

**public:**

vector**(){**

*// Dynamic Default constructor to the allocate 20 size to the newly formed array.*

array **=** **(int** **\*)** **(**malloc**(**20**\*sizeof(int)));**

**}**

**void** get\_details**();**

**void** display**();**

**class** vector **operator** **=** **(class** vector **&**temp**){**

size **=** temp**.**size**;**

**for(int** i **=** 0**;** i**<**temp**.**size**;** i**++){**

array**[**i**]** **=** temp**.**array**[**i**];**

**}**

**}**

**};**

**void** vector**::**get\_details**(){**

cout**<<"Enter the size of the array: ";**

cin**>>**size**;**

cout**<<"Enter the array elements: ";**

**for(int** i **=** 0**;** i**<**size**;** i**++){**

cin**>>**array**[**i**];**

**}**

cout**<<**endl**;**

**}**

**void** vector**::**display**(){**

cout**<<"The array elements are given: ";**

**for(int** i **=** 0**;** i**<**size**;** i**++){**

cout**<<**array**[**i**]<<" ";**

**}**

cout**<<**endl**;**

**}**

**int** main**(){**

**class** vector arrayA**,** arrayB**;**

arrayA**.**get\_details**();**

cout**<<"The original vector is given below: "<<**endl**;**

arrayA**.**display**();**

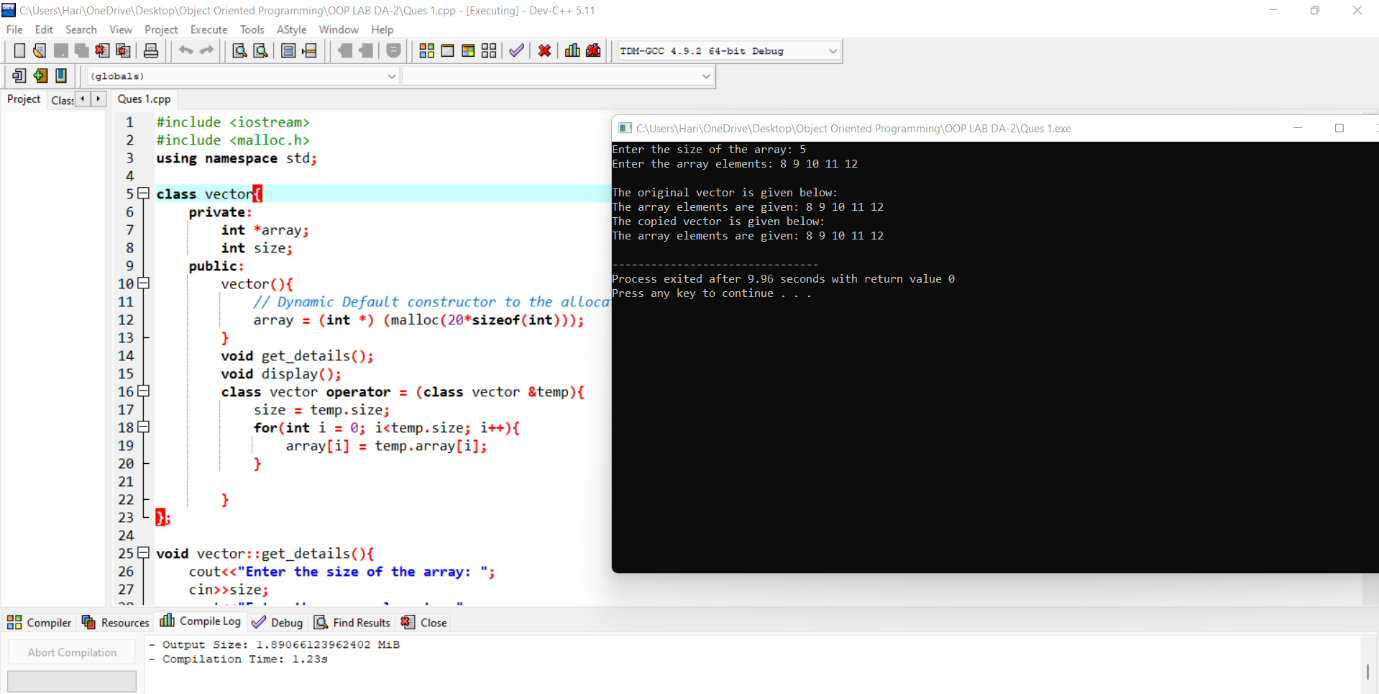
arrayB **=** arrayA**;**

cout**<<"The copied vector is given below: "<<**endl**;**

arrayB**.**display**();**

**return** 0**;**

**}**



Ques 3.

3. Develop an OOP to perform the addition, subtraction and multiplication of two matrices by overloading the +, - and \* operator. Define a constructor to allocate the memory space for the Matrix using dynamic memory allocation.

#include<iostream>

**using** **namespace** std**;**

*//Coded by Hari Krishna Shah*

**class** mat

**{**

**private:**

**int** s**[**10**][**10**];**

**int** r**,**c**;**

**public:**

**void** show**();**

mat **operator** **+(**mat**);**

mat **operator** **\*(**mat**);**

**void** read**();**

**};**

mat mat**::operator+(**mat obj**)**

**{**

mat t**;**

t**.**r**=**r**;**

t**.**c**=**c**;**

**for(int** i**=**0**;**i**<**t**.**r**;**i**++)**

**for(int** j**=**0**;**j**<**t**.**c**;**j**++){**

t**.**s**[**i**][**j**]=**s**[**i**][**j**]+**obj**.**s**[**i**][**j**];**

**}**

**return** t**;**

**}**

mat mat**::operator\*(**mat obj**)**

**{**

mat t**;**

t**.**r**=**r**;**

t**.**c**=**obj**.**c**;**

**for(int** i**=**0**;**i**<**t**.**r**;**i**++){**

**for(int** j**=**0**;**j**<**t**.**c**;**j**++)**

**{**

t**.**s**[**i**][**j**]=**0**;**

**for(int** k**=**0**;**k**<**c**;**k**++){**

t**.**s**[**i**][**j**]+=**s**[**i**][**k**]** **\*** obj**.**s**[**k**][**j**];**

**}**

**}**

**}**

**return** t**;**

**}**

**void** mat**::**read**()**

**{**

cout**<<"Enter Size of Matrix : \n";**

cin**>>**r**>>**c**;**

cout**<<"Enter the Elements of Matrix :\n";**

**for(int** i**=**0**;**i**<**r**;**i**++){**

**for(int** j**=**0**;**j**<**c**;**j**++){**

cin**>>**s**[**i**][**j**];**

**}**

**}**

**}**

**void** mat**::**show**()**

**{**

**for(int** i**=**0**;**i**<**r**;**i**++){**

**for(int** j**=**0**;**j**<**c**;**j**++){**

cout**<<**s**[**i**][**j**]<<"\t";**

**}**

cout**<<"\n";**

**}**

**}**

**int** main**()**

**{**

mat obj1 **,**obj2**,**obj3**;**

cout**<<"Enter First Matrix\n";**

obj1**.**read**();**

cout**<<**endl**;**

cout**<<"Enter Second Matrix\n";**

obj2**.**read**();**

obj3**=**obj1 **+** obj2**;**

cout**<<"Result After Addition of two Matrix\n";**

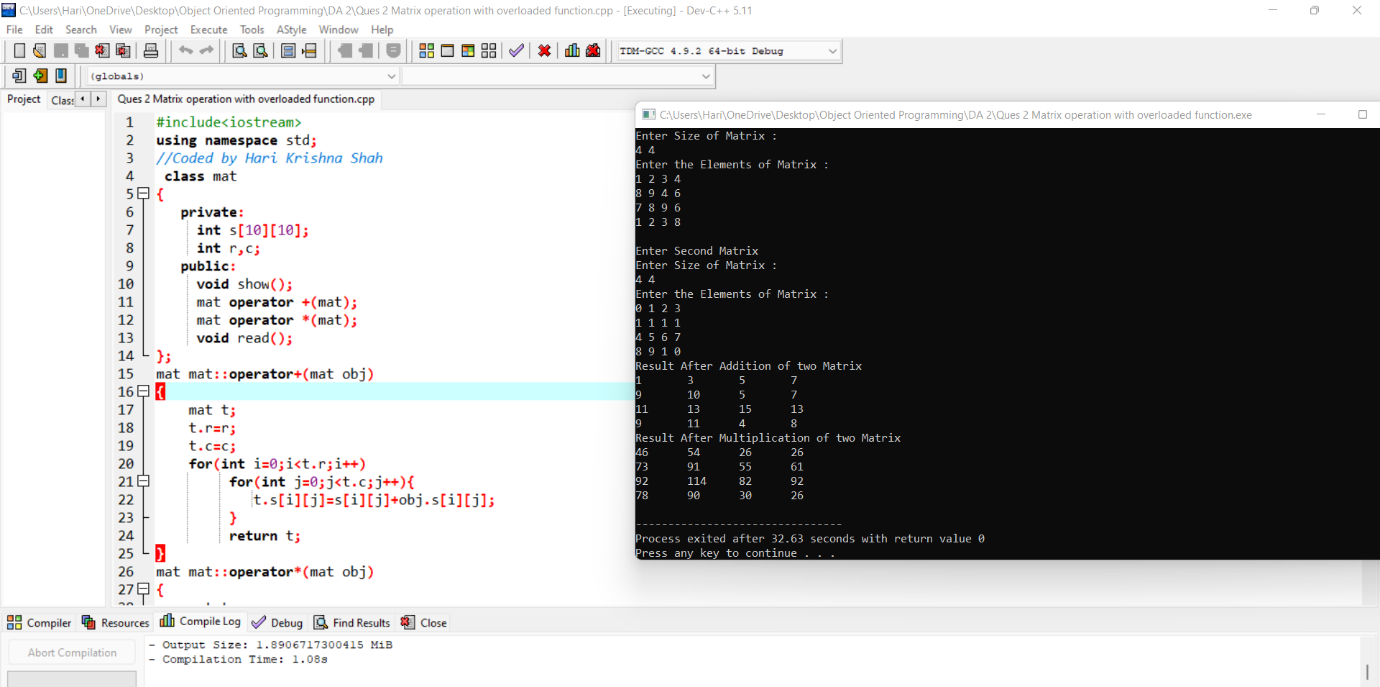
obj3**.**show**();**

obj3**=**obj1 **\*** obj2**;**

cout**<<"Result After Multiplication of two Matrix\n";**

obj3**.**show**();**

**}**

****