# Digital Assignment – 2

# Object Oriented Programming

**Name: Hari Krishna Shah**

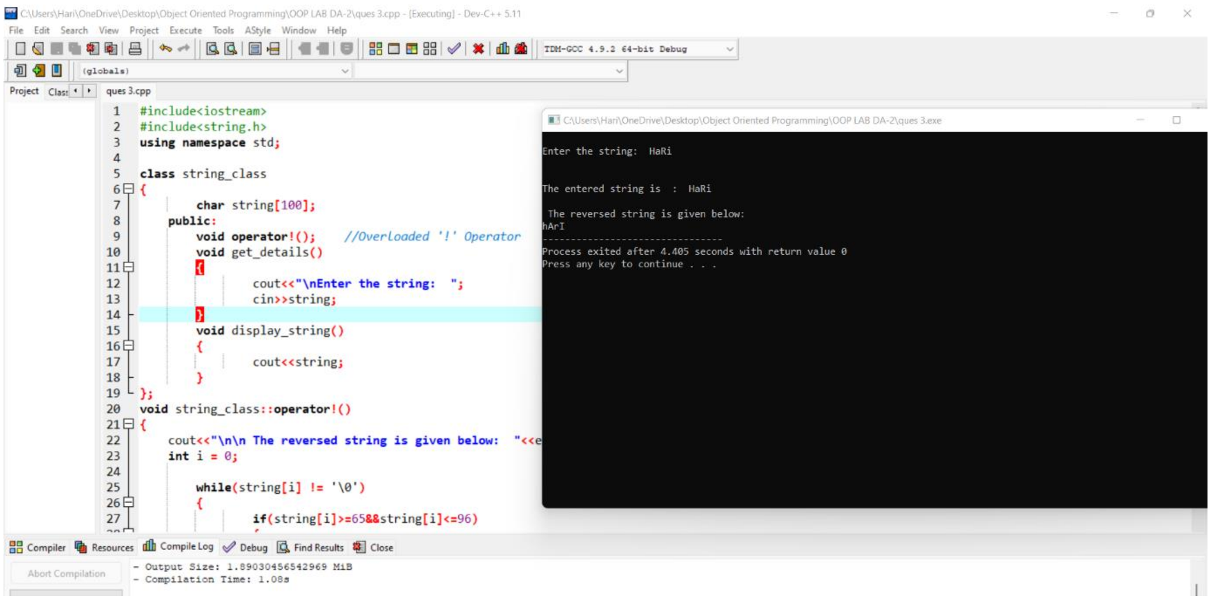
**VIT ID: 21BCS0167**

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;
}
```



C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\OOP LAB DA-2\ques 3.cpp - [Executing] - Dev-C++ 5.11

TDM-GCC 4.9.2 64-bit Debug

ques 3.cpp

```
#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:  "<<e
    int i = 0;

        while(string[i] != '\0')
        {
            if(string[i]>=65&&string[i]<=96)
```

C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\OOP LAB DA-2\ques 3.exe

```
Enter the string:  HaRi


The entered string is  :  HaRi

 The reversed string is given below:
hArI
--------------------------------
Process exited after 4.405 seconds with return value 0
Press any key to continue . . .
```

```
- Output Size: 1.89030456542969 MiB
- Compilation Time: 1.08s
```

Ques 2.

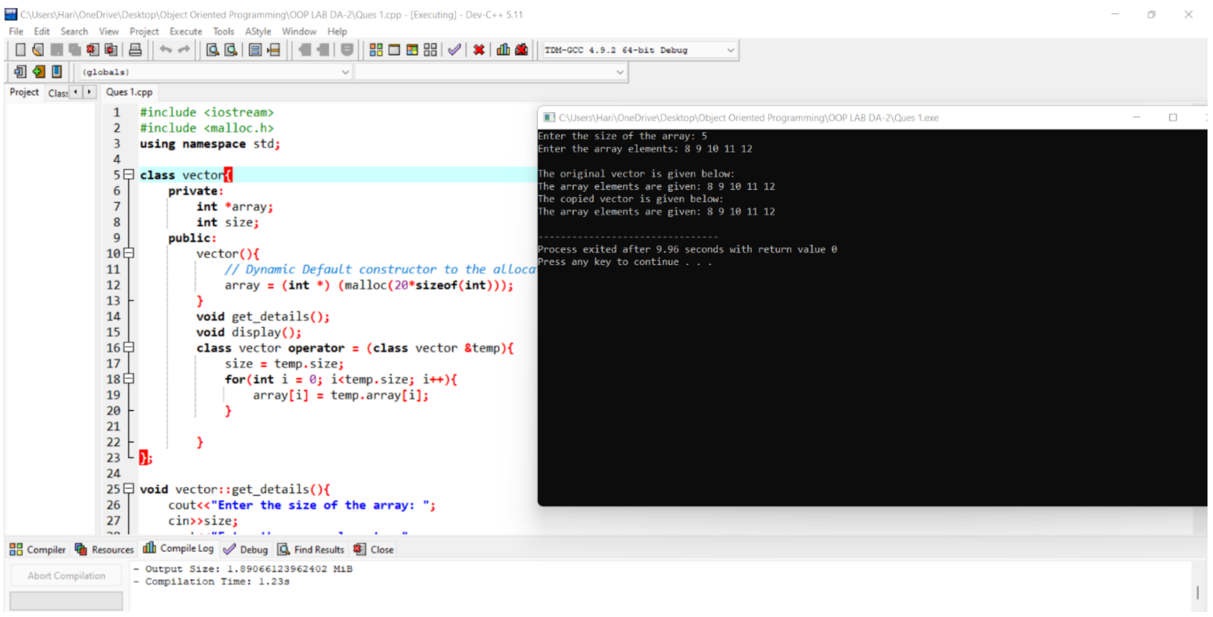
2. Develop an OOP to perform the assignment = operator overloading to assign one vector into another vector. Define a constructor to allocate the memory space for the vector using dynamic memory allocation. Note: Here, vector represents the single dimensional array which contains the set of values.

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;
}
```



C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\OOP LAB DA-2\Ques 1.cpp - [Executing] - Dev-C++ 5.11

TDM-GCC 4.9.2 64-bit Debug

Ques 1.cpp

```
#include <iostream>
#include <malloc.h>
using namespace std;

class vector{
    private:
        int *array;
        int size;
    public:
        vector(){
            // Dynamic Default constructor to the alloca
            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;
```

C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\OOP LAB DA-2\Ques 1.exe

```
Enter the size of the array: 5
Enter the array elements: 8 9 10 11 12

The original vector is given below:
The array elements are given: 8 9 10 11 12
The copied vector is given below:
The array elements are given: 8 9 10 11 12

--------------------------------
Process exited after 9.96 seconds with return value 0
Press any key to continue . . .
```

```
- Output Size: 1.89066123962402 MiB
- Compilation Time: 1.23s
```

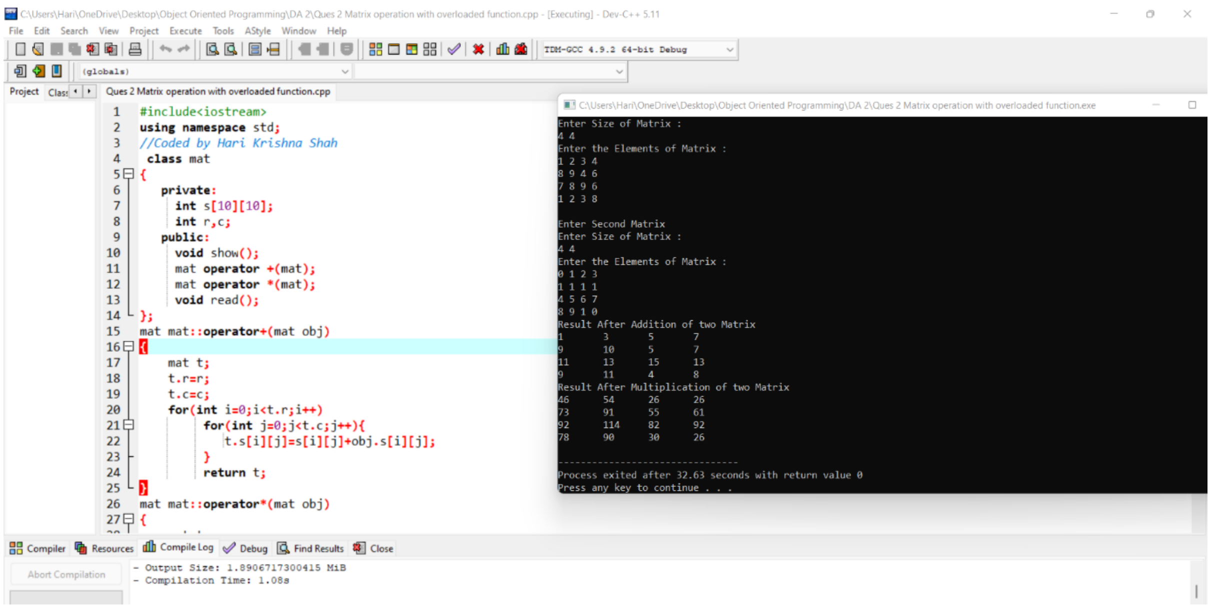
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Class
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation

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];
          }

   }

}
```

```cpp
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();
}
```



C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\DA 2\Ques 2 Matrix operation with overloaded function.cpp - [Executing] - Dev-C++ 5.11

TDM-GCC 4.9.2 64-bit Debug

Ques 2 Matrix operation with overloaded function.cpp

```cpp
#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)
{
```

C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\DA 2\Ques 2 Matrix operation with overloaded function.exe

```
Enter Size of Matrix :
4 4
Enter the Elements of Matrix :
1 2 3 4
8 9 4 6
7 8 9 6
1 2 3 8

Enter Second Matrix
Enter Size of Matrix :
4 4
Enter the Elements of Matrix :
0 1 2 3
1 1 1 1
4 5 6 7
8 9 1 0
Result After Addition of two Matrix
1       3       5       7
9       10      5       7
11      13      15      13
9       11      4       8
Result After Multiplication of two Matrix
46      54      26      26
73      91      55      61
92      114     82      92
78      90      30      26

--------------------------------
Process exited after 32.63 seconds with return value 0
Press any key to continue . . .
```

```
- Output Size: 1.8906717300415 MiB
- Compilation Time: 1.08s
```