B.Tech Computer Engineering

Vth Semester

Labs

Sub: OOPS Lab

Name: Syed Mohammad Raza
Roll No.: 18BCS056

Q1.To find number of vowels present in given character array using pointer arithmetic

```
#include <bits/stdc++.h>

int main()
{

    using namespace std;

    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    cout << "Syed Mohammad Raza(18BCS056)"
         << "\n";

    cout << "Program For Counting Vowels In The Given String Using
Pointer Arithmetic"
         << "\n";
    cout << "Enter Char. Array : ";
    char Char[1000];
    cin >> Char;
    char *str = Char;
    int counter = 0;
    while ((*str) != '\0')
    {
        char var = (*str);
        if (var == 'a' or var == 'e' or var == 'i' or var == 'o' or var
== 'u' or var == 'A' or var == 'E' or var == 'I' or var == 'O' or var
== 'U')
        {
            counter++;
        }
        str++;
    }

    cout << "Vowels In The Given String " << counter << "\n";

    return 0;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Syed Mohammad Raza(18BCS056)
Program For Counting Vowels In The Given String Using Pointer Arithmetic
Enter Char. Array : apple
Vowels In The Given String 2
```

Q2. Write a C++ program to print the given number in reverse order. Use functions with return type and without Return type for reversing the number.

```cpp
#include <bits/stdc++.h>

using namespace std;

int reverseNumberWithReturn(int n)
{
    int rev = 0;
    int rem;
    while (n != 0)
    {
        rem = n % 10;
        rev = (rev * 10) + rem;
        n /= 10;
    }
    return rev;
}

void reverseNumberWithoutReturn(int n)
{
    n = reverseNumberWithReturn(n);
    cout << "Reversed number is " << n << '\n';
}

int main()
{

    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    cout << "Enter the number : ";
    int no;
    cin >> no;
```

```
    cout << "Reversed number is " << reverseNumberWithReturn(no) <<
endl;
    reverseNumberWithoutReturn(no);

    return 0;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Syed Mohammad Raza(18BCS056)
Enter the number : 12345
Reversed number is 54321
Reversed number is 54321
```

Q3. Create a class for counting the number of objects created and destroyed within various block using constructor and destructor

```
#include <bits/stdc++.h>

using namespace std;

class counter
{
private:
    static int objectCreated;
    static int objectDestroyed;

public:
    counter()
    {
        objectCreated++;
    }

    ~counter()
    {
        objectDestroyed++;
    }
```

```
    static void info()
    {
        cout << " Objects created:" << objectCreated << endl;
        cout << " Objects destroyed: " << objectDestroyed << endl;
    }
};

int counter::objectCreated;
int counter::objectDestroyed;

int main()
{
    cout << "Syed Mohammad Raza(18BCS056)"
         << "\n";

    {
        counter *count1 = new counter();
        counter::info();
        counter *count2 = new counter();
        counter::info();
        counter *count3 = new counter();
        counter::info();

        delete count1;
        counter::info();
        delete count2;
        counter::info();
    }
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Syed Mohammad Raza(18BCS056)
 Objects created:1
 Objects destroyed: 0
 Objects created:2
 Objects destroyed: 0
 Objects created:3
 Objects destroyed: 0
 Objects created:3
 Objects destroyed: 1
 Objects created:3
 Objects destroyed: 2
```

Q4 To perform various arithmetic operations such as addition, subtraction, division , modulus and multiplication using inline function.

```cpp
#include <bits/stdc++.h>

using namespace std;

inline int add(int a, int b)
{
    return a + b;
}

inline int sub(int a, int b)
{
    return a - b;
}

inline int mult(int a, int b)
{
    return a * b;
}

inline float divsn(float a, float b)
{
    return a / b;
}
```

```
inline int mod(int a, int b)
{
    return a % b;
}

int main()
{
    int x, y;
    cout << "Enter Two Numbers: ";
    cin >> x >> y;
    cout << "Inline Addition: " << add(x, y) << "\n";
    cout << "Inline Subtraction: " << sub(x, y) << "\n";
    cout << "Inline Multiplication: " << mult(x, y) << "\n";
    cout << "Inline Division: " << divsn(x, y) << "\n";
    cout << "Inline Modulus: " << mod(x, y) << "\n";

    cout << "\n"
         << "Syed Mohammad Raza"
         << " 18BCS056"
         << "\n";
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Enter Two Numbers: 4 5
Inline Addition: 9
Inline Subtraction: -1
Inline Multiplication: 20
Inline Division: 0.8
Inline Modulus: 4

Syed Mohammad Raza 18BCS056
```

Q5.Write a Program To multiply two matrices dynamically

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
```

```
    int r, c, i, j, k;
    cout << "Enter Length Of Matrix 1"
         << "\n";
    cin >> r >> c;
    vector<vector<int>> matrix1(r, vector<int>(c));
    cout << "\n"
         << "Enter Elements Of Matrix 1:"
         << "\n";
    for (i = 0; i < r; ++i)
    {
        for (j = 0; j < c; ++j)
        {
            cout << "Enter Elements "
                 << " : ";
            cin >> matrix1[i][j];
        }
    }
    cout << "Enter Size Of Matrix 2"
         << "\n";
    cin >> r >> c;
    vector<vector<int>> matrix2(r, vector<int>(c));
    cout << "\n"
         << "Enter Elements Of Matrix 2:"
         << "\n";
    for (i = 0; i < r; ++i)
    {
        for (j = 0; j < c; ++j)
        {
            cout << "Enter Elements "
                 << " : ";
            cin >> matrix2[i][j];
        }
    }
    vector<vector<int>> matrixFinal(matrix1.size(),
vector<int>(matrix2[0].size()));
    for (i = 0; i < matrix1.size(); ++i)
        for (j = 0; j < matrix2[0].size(); ++j)
            for (k = 0; k < matrix1[0].size(); ++k)
            {
                matrixFinal[i][j] += matrix1[i][k] * matrix2[k][j];
            }
    cout << "Desired Product : "
         << "\n";
```

```
    for (i = 0; i < matrixFinal.size(); ++i)
    {
        for (j = 0; j < matrixFinal[0].size(); ++j)
        {
            cout << matrixFinal[i][j] << " ";
        }
        cout << "\n";
    }

    cout << "Syed Mohammad Raza"
         << " 18BCS056"
         << "\n";
    return 0;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Enter Length Of Matrix 1
3 3

Enter Elements Of Matrix 1:
Enter Elements  : 1
Enter Elements  : 2
Enter Elements  : 3
Enter Elements  : 4
Enter Elements  : 5
Enter Elements  : 6
Enter Elements  : 7
Enter Elements  : 8
Enter Elements  : 9
Enter Size Of Matrix 2
3 3

Enter Elements Of Matrix 2:
Enter Elements  : 10
Enter Elements  : 11
Enter Elements  : 12
Enter Elements  : 13
Enter Elements  : 14
Enter Elements  : 15
Enter Elements  : 16
Enter Elements  : 17
Enter Elements  : 18
Desired Product :
84 90 96
201 216 231
318 342 366
Syed Mohammad Raza 18BCS056
```

Q6. Create 3 Objects of a Class named pntr_obj with data members such as roll_no & name. Create a member function set_data() for setting the data values and print() member function to print which object has invoked it using 'this' pointer.

```cpp
#include <bits/stdc++.h>
using namespace std;
```

```
class ptr_obj
{
public:
    string rno, name;
    void set_data()
    {
        cout << "Enter Name:";
        cin >> this->name;
        cout << "Enter Roll no: ";
        cin >> this->rno;
    }
    void print()
    {
        cout << "Name - " << this->name << "\nRoll No - " << this->rno
<< endl;
    }
};
int main()
{
    cout << "Syed Mohammad Raza\n18BCS056" << endl;
    ptr_obj *ob1 = new ptr_obj(), *ob2 = new ptr_obj(), *ob3 = new
ptr_obj();
    ob1->set_data();
    ob2->set_data();
    ob3->set_data();
    ob1->print();
    return 0;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Syed Mohammad Raza
18BCS056
Enter Name:joey
Enter Roll no: 1
Enter Name:luigi
Enter Roll no: 2
Enter Name:salvatore
Enter Roll no: 3
Name - joey
Roll No - 1
```

Q7. Implement Virtual function (polymorphism) by creating a base class c_polygon which has virtual function area(). Two classes c_rectangle and c_ triangle derived from c_polygon and they have area() to calculate and return the area of rectangle & triangle respectively.

```cpp
#include <bits/stdc++.h>
using namespace std;
class c_polygon
{
public:
    int h, b;
    void set_values()
    {
        cout << "Enter Height and Breath : ";
        cin >> h >> b;
    }
    virtual float area()
    {
        return 0;
    }
};
class c_rectangle : public c_polygon
{
public:
    float area()
    {
        return h * b;
    }
};
```

```cpp
class c_triangle : public c_polygon
{
public:
    float area()
    {
        return h * b * 0.5;
    }
};
int main()
{
    cout << "Syed Mohammad Raza \n18BCS056" << endl;
    c_triangle trg;
    c_rectangle rect;
    rect.set_values();
    trg.set_values();
    cout << "Rectangle area : " << rect.area() << endl;
    cout << "Trianle area : " << trg.area() << endl;
    return 0;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Syed Mohammad Raza
18BCS056
Enter Height and Breath : 12 9
Enter Height and Breath : 10 5
Rectangle area : 108
Trianle area : 25
```

Q8. To demonstrate Banking Operation using class such as -Open an account(SB/Current account) -Deposit into account. -Withdrawal from account with min. balance condition. -Balance inquiry. -Customer Contact update etc. -Exit.

```
#include <bits/stdc++.h>
using namespace std;
const int min_bal = 500;
class Bank
{
public:
    int bal, acc_id;
    string contact_no, name;
    Bank(int id)
    {
        cout << "Enter Name : ";
        cin >> name;
        cout << "Enter Contact no : ";
        cin >> contact_no;
        bal = 0;
        acc_id = id;
        cout << "Your account ID is " << acc_id << endl;
        deposit();
    }
    void balance_enquiry()
    {
        cout << "Current Amount is : " << bal << endl;
    }
    void deposit()
    {
        cout << "Enter amount to be deposited : ";
        int x;
        cin >> x;
        while (bal + x < min_bal)
            cout << "Amount less than min\n";
        cout << "Enter amount again : ";
        cin >> x;
        bal += x;
    }
    void withdraw()
    {
        cout << "Enter withdrawl amount : ";
        int x;
```

```
        cin >> x;
        if (bal - x < min_bal)
        {
            cout << "Account Bal min" << endl;
            return;
            bal -= x;
            cout << "Amount Withdrawn Successfully" << endl;
            balance_enquiry();
        }
    }
    void customer_update()
    {
        cout << "Update Name (Y/N): ";
        char ch;
        cin >> ch;
        if (ch == 'Y')
        {
            cout << "Enter new name";
            cin >> name;
            cout << "\nUpdate Contact no.(Y/N): ";
            cin >> ch;
            if (ch == 'Y')
            {
                cout << "Enter new contact no";
                cin >> contact_no;
            }
        }
    }
};
int main()
{
     cout << "Syed Mohammad Raza \n18BCS056" << endl;
    vector<Bank> v;
    while (true)
    {
        cout << "1. Open Account\n 2.  Deposit into Account\n 3.
Withdraw From Account\n 4. Balance Enquiry\n 5. Customer Contact
Update\n 6. Exit\n";

        int ch;
        cin >> ch;
        if (ch == 1)
        {
```

```
            int id = v.size();
            cout << "Your new Account id: " << id << endl;
            Bank acc(id);
            v.push_back(acc);
        }
        else if (ch == 2)
        {
            cout << "Enter Account ID: ";
            int id;
            cin >> id;
            if (id >= v.size())
                cout << "Invalid ID" << endl;
            else
                v[id].deposit();
        }
        else if (ch == 3)
        {
            cout << "Enter Account ID: ";
            int id;
            cin >> id;
            if (id >= v.size())
                cout << "Invalid ID" << endl;
            else
                v[id].withdraw();
        }
        else if (ch == 4)
        {
            cout << "Enter Account ID: ";
            int id;
            cin >> id;
            if (id >= v.size())
                cout << "Invalid ID" << endl;
            else
                v[id].balance_enquiry();
        }
        else if (ch == 5)
        {
            cout << "Enter Account ID: ";
            int id;
            cin >> id;
            if (id >= v.size())
                cout << "Invalid ID" << endl;
            else
```

```
                v[id].customer_update();
        }
        else
            break;
    }
    return 0;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Syed Mohammad Raza
18BCS056
1. Open Account
 2.  Deposit into Account
 3. Withdraw From Account
 4. Balance Enquiry
 5. Customer Contact Update
 6. Exit
1
Your new Account id: 0
Enter Name : Mohd. Raza
Enter Contact no : Your account ID is 0
Enter amount to be deposited : 1000
Enter amount again : 1000
1. Open Account
 2.  Deposit into Account
 3. Withdraw From Account
 4. Balance Enquiry
 5. Customer Contact Update
 6. Exit
4
Enter Account ID: 0
Current Amount is : 1000
1. Open Account
 2.  Deposit into Account
 3. Withdraw From Account
 4. Balance Enquiry
 5. Customer Contact Update
 6. Exit
```

Q9.To count the number of persons inside a bank by increasing count whenever a person enters a bank using an increment operator(++) overloading function and decrease the count whenever a person leaves the bank using a decrement (--) operator overloading function inside a class.

```
#include <bits/stdc++.h>
using namespace std;

class Bank
{
public:
    int p;
    Bank()
    {
        p = 0;
    }

    void operator++(int)
    {
        p++;
    }
    void operator--(int)
    {
        p--;
    }
    void print()
    {
        cout << "Number of people in the Bank are: " << p << endl;
    }
};
int main()
{
    cout << "Syed Mohammad Raza \n18BCS056" << endl;
    Bank B1;
    B1.print();
    B1++;
    B1++;
    B1.print();
    B1--;
    B1.print();
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Syed Mohammad Raza
18BCS056
Number of people in the Bank are: 0
Number of people in the Bank are: 2
Number of people in the Bank are: 1
```

Q 10.To create two objects of a class called company and add their data members using an operator overloaded function for '+' operator and '-' operator.

```
#include <bits/stdc++.h>
using namespace std;
class Company
{
public:
    int e;
    Company(int x = 0) { e = x; }

    Company operator+(Company x)
    {
        Company c(e + x.e);
        return c;
    }
    Company operator-(Company x)
    {
        Company c(e - x.e);
        return c;
    }
    void print()
    {
        cout << e << endl;
    }
};
int main()
{
    cout << "Syed Mohammad Raza \n18BCS056" << endl;
    Company c1(11), c2(220);
    auto c = c1 + c2;
    c.print();
    c = c1 - c2;
```

```
    c.print();
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Syed Mohammad Raza
18BCS056
231
-209
```

Q 11.To create a class matrix and overload +,- operator to perform matrix addition and subtraction

```
#include <bits/stdc++.h>
using namespace std;
class Matrix
{
private:
    int r;
    int c;
    int mat[200][200];

public:
    void input()
    {
        cout << "Enter number of rows: ";
        cin >> r;
        cout << "Enter number of column : ";
        cin >> c;
        cout << "Enter the matrix : " << endl;
        for (int i = 0; i < r; i++)
        {
            for (int j = 0; j < c; j++)
            {
                cin >> mat[i][j];
            }
        }
    }
    void display()
    {
        for (int i = 0; i < r; i++)
        {
```

```
            for (int j = 0; j < c; j++)
            {
                cout << mat[i][j] << " ";
            }
            cout << endl;
        }
    }
    Matrix operator+(Matrix obj)
    {
        if ((this->r) != obj.r || (this->c != obj.c))
        {
            cout << "Dimensions don't match , So elementwise operation
cannot take place " << endl;
            Matrix temp;
            return temp;
        }
        else
        {
            Matrix temp;
            for (int i = 0; i < this->r; i++)
            {
                for (int j = 0; j < this->c; j++)
                {
                    temp.mat[i][j] = this->mat[i][j] + obj.mat[i][j];
                }
            }
            temp.r = this->r;
            temp.c = this->c;
            return temp;
        }
    }
    Matrix operator-(Matrix obj)
    {
        if ((this->r) != obj.r || (this->c != obj.c))
        {
            cout << "Dimensions don't match , So elementwise operation
cannot take place " << endl;
            Matrix temp;
            return temp;
        }
        else
        {
            Matrix temp;
```

```
            for (int i = 0; i < this->r; i++)
            {
                for (int j = 0; j < this->c; j++)
                {
                    temp.mat[i][j] = this->mat[i][j] - obj.mat[i][j];
                }
            }
            temp.r = this->r;
            temp.c = this->c;
            return temp;
        }
    }
};
int main()
{
    cout << "Syed Mohammad Raza \n18BCS056" << endl;
    Matrix mat1;
    Matrix mat2;
    mat1.input();
    mat2.input();
    Matrix mat3 = mat1 + mat2;
    Matrix mat4 = mat1 - mat2;
    cout << "Addition of two matrices : " << endl;
    mat3.display();
    cout << "Subtraction of two matrices: " << endl;
    mat4.display();
    return 0;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Syed Mohammad Raza
18BCS056
Enter number of rows: 3
Enter number of column : 3
Enter the matrix :
1 2 3
4 5 6
7 8 9
Enter number of rows: 3
Enter number of column : 3
Enter the matrix :
10 11 12
13 14 15
16 17 18
Addition of two matrices :
11 13 15
17 19 21
23 25 27
Subtraction of two matrices:
-9 -9 -9
-9 -9 -9
-9 -9 -9
```

Q 12.Write a program that uses a function template called min to determine the smaller of two arguments. The program should work for integers, characters and floating-point number as arguments to this function.

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
template <typename T>
T myMin(T x, T y)
{

    if (x == y)
    {
        cout << "Same Same Same!!!!" << endl;
    }
```

```
    return min(x, y);
}
int main()
{
    int n1, n2;
    cout << "Enter Two Numbers: ";
    cin >> n1 >> n2;
    cout << myMin<int>(n1, n2) << endl;
    char c1, c2;
    cout << "Enter Two Characters: ";
    cin >> c1 >> c2;
    cout << myMin<char>(c1, c2) << endl;
    float f1, f2;
    cout << "Enter Two Float Numbers: ";
    cin >> f1 >> f2;
    cout << myMin<float>(f1, f2) << endl;

    cout << "Syed Mohammad Raza" << endl;
    cout << "18BCS056" << endl;
    return 0;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Enter Two Numbers: 3 4
3
Enter Two Characters: a z
a
Enter Two Float Numbers: 2.0 3.9
2
Syed Mohammad Raza
18BCS056
```

Q 13.Write a program to explain class template by creating a template T for a class named pair having two data members of type T which are inputted by a constructor and a member function get-max() return the greatest of two numbers to main.

```
#include <bits/stdc++.h>
using namespace std;
```

```cpp
typedef long long int ll;
template <typename T>
class Pair
{
public:
    T x;
    T y;
    Pair(T x, T y)
    {
        this->x = x;
        this->y = y;
    }
    T getMax()
    {
        if (x == y)
        {
            cout << "Same Same Same!!!!" << endl;
        }

        return max(x, y);
    }
};
int main()
{
    int n1, n2;
    cout << "Enter Two Numbers: ";
    cin >> n1 >> n2;
    Pair<int> pair1 = Pair<int>(n1, n2);
    cout << pair1.getMax() << endl;

    char c1, c2;
    cout << "Enter Two Characters: ";
    cin >> c1 >> c2;
    Pair<char> pair2 = Pair<char>(c1, c2);
    cout << pair2.getMax() << endl;

    float f1, f2;
    cout << "Enter Two Float Numbers: ";
    cin >> f1 >> f2;
    Pair<float> pair3 = Pair<float>(f1, f2);
    cout << pair3.getMax() << endl;

    cout << "Syed Mohammad Raza" << endl;
```

```
    cout << "18BCS056" << endl;
    return 0;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Enter Two Numbers: 3 4
4
Enter Two Characters: a z
z
Enter Two Float Numbers: 2.0 3.999
3.999
Syed Mohammad Raza
18BCS056
```

Q 14.Write a program in C++ to overload Cin, Cout stream operators to input and display objects of a class student which contains student details like name, roll no., class, age.
Eg: class Student{.....}
void main()
{ student st;
Cin>>st;
Cout<<st;
}

```
#include <bits/stdc++.h>
using namespace std;
class Student
{
private:
    int rollNo;
    string name;
    int standard;

public:
    Student(int rollNo, string name, int standard)
    {
        this->rollNo = rollNo;
        this->name = name;
```

```
        this->standard = standard;
    }
    friend ostream &operator<<(ostream &out, Student &s);
};
ostream &operator<<(ostream &out, Student &s)
{
    cout << "Roll No: " << s.rollNo << endl;
    cout << "Name: " << s.name << endl;
    cout << "Class: " << s.standard << endl;
    return out;
}
int main()
{
    Student s1(1, "Syed Mohammad Raza", 56);
    Student s2(2, "Md Ruhulamin Khan", 67);
    Student s3(3, "Dhruv Dua", 89);
    cout << s1;
    cout << s2;
    cout << s3;
}
```

```
C:\Users\raza9\Desktop\Labs\OOPS Lab>a.exe
Roll No: 1
Name: Syed Mohammad Raza
Class: 56
Roll No: 2
Name: Md Ruhulamin Khan
Class: 67
Roll No: 3
Name: Dhruv Dua
Class: 89
Syed Mohammad Raza
18BCS056
```