CSE Assignment (C++)

Problem 1

Question:

> *Create a class Matrix to store a m x n matrix. Include necessary constructors and functions toinitialize and display the matrix. Using friend function, find the dot product of two input matrices*

Code:

```
#include <iostream>

using namespace std;

class Matrix
{
    private:

        int grid[3][3];

    public:

        Matrix()
        {
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    grid[i][j] = 0;
                }
            }
        }

        Matrix(int (&cp)[3][3])
        {
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    grid[i][j] = cp[i][j];
                }
            }
        }

        void dispMatrix()
        {
            for (int i = 0; i < 3; i++)
            {
                cout << endl;
```

```
                for (int j = 0; j < 3; j++)
                {
                    cout << grid[i][j] << " ";
                }
            }
        }

        friend void dotP(Matrix &op1, Matrix &op2);
};

void dotP(Matrix &op1, Matrix &op2)
{
    int prod[3][3];

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            prod[i][j] = 0;

            for (int k = 0; k < 3; k++)
            {
                prod[i][j] += op1.grid[i][k] * op2.grid[k][j];
            }
        }
    }

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            op1.grid[i][j] = prod[i][j];
        }
    }
}

int main()
{
    int arr[3][3] = {{1, 2, 3}, {1, 2, 3}, {1, 2, 3}};

    Matrix m1(arr), m2(arr);

    dotP(m1, m2);

    m1.dispMatrix();

    cout << endl;

    return 0;
}
```

Output:

```
6 12 18
6 12 18
6 12 18
```

Problem 2

Question:

> *Create two classes IntArray to store the set of integer numbers and FloatArray to store decimalnumbers. Add a member function read() for both classes for reading inputs. Create two objects 'x' for IntArray and 'y' for FloatArray. Read the inputs for x and y. Using a friend function maxmin(x,y), display the maximum and minimum among the set of integers and decimal numbers*

Code:

```
#include <iostream>

using namespace std;

class IntArray;
class FloatArray;

class IntArray
{
private:
    int arr[5];

public:

    IntArray();
    IntArray(int (&cp)[5]);
    int read(int index) const;
    friend void minmax(IntArray &x, FloatArray &y);
};

class FloatArray
{
private:
    float arr[5];

public:

    FloatArray();
    FloatArray(float (&cp)[5]);
    float read(int index) const;
```

```
    friend void minmax(IntArray &x, FloatArray &y);
};

IntArray::IntArray()
{
    for (int i = 0; i < 5; i++)
    {
        arr[i] = 0;
    }
}

IntArray::IntArray(int (&cp)[5])
{
    for (int i = 0; i < 5; i++)
    {
        arr[i] = cp[i];
    }
}

int IntArray::read(int index) const
{
    return arr[index];
}

FloatArray::FloatArray()
{
    for (int i = 0; i < 5; i++)
    {
        arr[i] = 0;
    }
}

FloatArray::FloatArray(float (&cp)[5])
{
    for (int i = 0; i < 5; i++)
    {
        arr[i] = cp[i];
    }
}

float FloatArray::read(int index) const
{
    return arr[index];
}

void minmax(IntArray &x, FloatArray &y)
{
    float min = 6;
    float max = 0;

    for (int i = 0; i < 5; i++)
    {
```

```
        if (min > x.read(i))
        {
            min = x.arr[i];

            if (min > y.read(i))
            {
                min = y.arr[i];
            }
        }

        if (max < x.read(i))
        {
            max = x.arr[i];

            if (max < y.read(i))
            {
                max = y.arr[i];
            }
        }
    }

    cout << "Minimum : " << min << endl;
    cout << "Maximum : " << max << endl;
}

int main()
{
    int iarr[] = {1, 2, 3, 4, 5};
    float farr[] = {0.5, 1.5, 2.5, 3.5, 4.5};
    IntArray x(iarr);
    FloatArray y(farr);

    minmax(x, y);

    return 0;
}
```

Output:

```
Minimum : 0.5
Maximum : 5
```

Problem 3

Question:

> *Develop a C++ program to using pure virtual function to find area of different shapes*

Code:

```cpp
#include <iostream>

using namespace std;

class Areator;
class Circle;
class Triangle;
class Rectangle;

class Areator
{
public:
    virtual void computeArea() = 0;
};

class Circle : private Areator
{
private:
    float radius;
    float area;

public:
    Circle();
    Circle(float r);

    virtual void computeArea();
    friend void printArea(Circle &obj);
};

class Triangle : private Areator
{
private:
    float base;
    float height;
    float area;

public:
    Triangle();
    Triangle(float b, float h);

    virtual void computeArea();
    friend void printArea(Triangle &obj);
};

class Rectangle : private Areator
{
private:
    float length;
    float breadth;
    float area;
```

```
public:
    Rectangle();
    Rectangle(float l, float b);

    virtual void computeArea();
    friend void printArea(Rectangle &obj);
};

Circle::Circle()
{
    radius = 0;
    area = 0;
}

Circle::Circle(float r)
{
    radius = r;
    area = 0;
}

void Circle::computeArea()
{
    area = (3.14 * radius * radius);
}

void printArea(Circle &obj)
{
    cout << "The area of the circle is : " << obj.area;
}

Triangle::Triangle()
{
    base = 0;
    height = 0;
    area = 0;
}

Triangle::Triangle(float b, float h)
{
    base = b;
    height = h;
    area = 0;
}

void Triangle::computeArea()
{
    area = (0.5 * base * height);
}

void printArea(Triangle &obj)
{
    cout << "The area of the triangle is : " << obj.area;
```

```
}

Rectangle::Rectangle()
{
    length = 0;
    breadth = 0;
    area = 0;
}

Rectangle::Rectangle(float l, float b)
{
    length = l;
    breadth = b;
    area = 0;
}

void Rectangle::computeArea()
{
    area = (length * breadth);
}

void printArea(Rectangle &obj)
{
    cout << "The area of the rectangle is : " << obj.area;
}

int main(int argc, char const *argv[])
{
    Circle circle(5);
    Triangle triangle(5, 10);
    Rectangle rectangle(5, 5);

    circle.computeArea();
    triangle.computeArea();
    rectangle.computeArea();

    printArea(circle);
    cout << endl;
    printArea(triangle);
    cout << endl;
    printArea(rectangle);
    cout << endl;

    return 0;
}
```

Output:

```
The area of the circle is : 78.5
The area of the triangle is : 25
The area of the rectangle is : 25
```

Problem 4

Question:

> *Write a program to convert the text file contents to Upper-case & write the contents into another file*

Code:

```
#include <fstream>
#include <string.h>

using namespace std;

int main(int argc, char const *argv[])
{
    char symbol;
    fstream fead;
    fstream frite;

    fead.open("lowercase.txt", ios::in);
    frite.open("UPPERCASE.txt", ios::out);

    if (!fead || !frite)
    {
        return -1;
    }

    while (!fead.eof())
    {
        fead.get(symbol);
        frite << (char)toupper(symbol);
    }

    fead.close();
    frite.close();

    return 0;
}
```

Output:

```
JUST SOME RANDOM CSE STUFFF
```

Problem 5

Question:

> *Write a program to store Student objects in a file, search for a student and display the student's details*

Code:

```
#include <iostream>
#include <fstream>
using namespace std;

class Student
{
private:
    int roll_no;
    string name;

public:
    void getdata()
    {
        cout << "Enter roll no : ";
        cin >> roll_no;
        cout << "Enter name : ";
        cin >> name;
        cout << endl;
    }

    void display()
    {
        cout << endl
             << "Roll no : " << roll_no;
        cout << endl
             << "Name : " << name;
    }
};

int main()
{
    int offset = sizeof(Student) * 2;

    ofstream of("student.dat", ios::binary);

    if (!of)
    {
        cout << "Cannot open file!" << endl;

        return 1;
    }

    Student stu_w[3];

    for (int i = 0; i < 3; i++)
    {
        stu_w[i].getdata();
    }

    for (int i = 0; i < 3; i++)
```

```
    {
        of.write((char *)&stu_w[i], sizeof(Student));
    }

    of.close();

    if (!of.good())
    {
        cout << "Error occurred at writing time!" << endl;

        return 1;
    }

    ifstream ifs("student.dat", ios::binary);

    if (!ifs)
    {
        cout << "Cannot open file!" << endl;

        return 1;
    }

    Student stu_r;

    ifs.seekg(offset, ios_base::beg);
    ifs.read((char *)&stu_r, sizeof(Student));
    ifs.close();

    if (!ifs.good())
    {
        cout << "Error occurred at reading time!" << endl;

        return 1;
    }

    cout << "Student's Details :" << endl;

    stu_r.display();
    cout << endl;

    return 0;
}
```

Output:

```
Enter roll no : 1
Enter name : BC

Enter roll no : 2
Enter name : VJ

Enter roll no : 3
```

```
Enter name : JS

Student's Details :

Roll no : 3
Name : JS
```