Saroj Dahal 191725

Q1. Write a complete definition of the class Rational with its method defined outside. The program shoulda.Define Rational object called xb.Assign values to data of xc.Compare num and den and display minimum valued.Define another Rational object called ye.Multiply two rational numbersf.Check if you can print num and den of x from main. Discuss the reason behind your observationg.If you were not able to access num and den of x from outside, what should you do so that you can access them from anywhere?

```
#include <iostream>

using namespace std;

class Rational
{

public:
   int deno, num;

   void setData();

   void minValue();
   void multiplyRational(Rational x, Rational y);
};

void Rational ::setData()
{

   cout << "Enter Numerator and Denominator : ";
   cin >> num >> deno;
}
void Rational ::minValue()
{
   (deno < num) ? cout << deno << " is minimum" : cout << num << " is minimum";
}

void Rational::multiplyRational(Rational x, Rational y)
{
   cout << endl
        << " Multiplication is : " << (x.num * y.num) << "/" << (x.deno * y.deno);
}

int main()
{

   Rational x, y, obj;
   x.setData();
```

```
    x.minValue();

    y.setData();
    y.minValue();

    obj.multiplyRational(x, y);

    return 1;
}
```

Q2. Write a program to define class Rectangle with length and breadth as private members and appropriate member functions to read data members, calculate and display the area.Define member functions outside class.

```
#include <iostream>

using namespace std;
class Rectangle
{

private:
    int length, breadth;

public:
    void setData();
    void displayData();
    void findArea();
};

void Rectangle ::setData()
{

    cout << "Enter rectangle length and breadth : ";
    cin >> length >> breadth;
}

void Rectangle ::displayData()
{

    cout << "Given Length : " << length << " Breadth : " << breadth;
}

void Rectangle ::findArea()
{
```

```
    cout << "Area of rectangle is : " << (length * breadth);
}

int main()
{

    Rectangle rectangle;

    rectangle.setData();
    rectangle.displayData();
    rectangle.findArea();

    return 1;
}
```

Q3. Define a class named Complex with data members real and img. Use appropriate member function of the class which accepts two objects of the class and addsthem.

```
#include <iostream>

using namespace std;
class Complex
{

public:
    int real, img;

    void getData()
    {

        cout << "Enter real and imaginary value : ";
        cin >> real >> img;
    }

    void addData(Complex c1, Complex c2)
    {

        cout << " The sum of two complex number is : " << (c1.real + c2.real) << " " << (c1.img +
c2.img) << endl;
    }
};

int main()
{
    Complex c1, c2, obj;
```

```
    c1.getData();
    c2.getData();
    obj.addData(c1, c2);

    return 1;
}
```

Q4. Define a class named Complex with data members real and img. Use friend function to add two complex numbers.

```
#include <iostream>

using namespace std;
class Complex
{

private:
    int real, img;

public:
    void setData()
    {
        cout << "Enter real and imaginary number : ";
        cin >> real >> img;
    }
    friend void addComplex(Complex c1, Complex c2);
};

void addComplex(Complex c1, Complex c2)
{
    cout << "Sum of complex number is : " << (c1.real + c2.real) << " " << (c1.img + c2.img);
}

int main()
{
    Complex c1, c2;
    c1.setData();
    c2.setData();
    addComplex(c1, c2);
    return 1;
}
```

Saroj Dahal 191725

Q5. Write a program to define class Distance with data members feet and inches of appropriate type. Define member function of the class which accepts two objects of the class and adds them.

```
#include <iostream>

using namespace std;

class Distance
{

private:
   int feet, inches;

public:
   void setData()
   {

      cout << "Enter feet and inches : ";
      cin >> feet >> inches;
   }

   void addData(Distance d1, Distance d2)
   {
      int totalFeet = d1.feet + d2.feet + (d1.inches % 12) + (d2.inches % 12);
      int totalInches = (d1.inches / 12) + (d2.inches / 12.0);
      cout << "Sum :" << totalFeet << " Foot and " << totalInches << " Inches ";
   }
};

int main()
{
   Distance d1, d2, obj;
   d1.setData();
   d2.setData();
   obj.addData(d1, d2);

   return 1;
}
```

Q6. Write a program to define class Distance with data members feet and inches of appropriate type. Define friend functionwhich accepts two objects of Distanceand adds them.

```
#include <iostream>
```

```
using namespace std;

class Distance
{

private:
   int feet, inches;

public:
   void setData()
   {

      cout << "Enter feet and inches : ";
      cin >> feet >> inches;
   }

   friend void addData(Distance d1, Distance d2);
};

void addData(Distance d1, Distance d2)
{
   int totalFeet = d1.feet + d2.feet + (d1.inches % 12) + (d2.inches % 12);
   int totalInches = (d1.inches / 12) + (d2.inches / 12.0);
   cout << "Sum :" << totalFeet << " Foot and " << totalInches << " Inches ";
}

int main()
{
   Distance d1, d2, obj;
   d1.setData();
   d2.setData();
   addData(d1, d2);

   return 1;
}
```

Q7. Write a program to swap private data of two classes using friend function.

```
#include <iostream>

using namespace std;

class Second;
class First
```

```
{
   int a;

public:
   friend void swap(First, Second);
};

class Second
{
   int b;

public:
   friend void swap(First, Second);
};

void swap(First f, Second s)
{
   int temp;
   f.a = 10;
   s.b = 20;
   cout << "\nThe default value of";
   cout << "\na=" << f.a;
   cout << "\nb=" << s.b;

   //swaping
   temp = f.a;
   f.a = s.b;
   s.b = temp;
   cout << "\nNow the changed value of";
   cout << "\na=" << f.a;
   cout << "\nb=" << s.b;
}
int main()
{
   First f;
   Second s;
   swap(f, s);
   return 1;
}
```

Q8. Create classes called Class1,Class2, Class3, Class4with each having one private member. Add member function to set a value (say setValue) on each class. Add one more function max() that is friendly to allclasses, max() function should compare four private member of fourclasses and show maximum among them. Create one-one object of each class and set a value on them. Display the maximum number among them.

Saroj Dahal 191725

```
#include <iostream>

using namespace std;

class Class2;
class Class3;
class Class4;

class Class1
{
private:
    int val;

public:
    void setValue(int value)
    {
        val = value;
    }

    friend void max(Class1 c1, Class2 c2, Class3 c3, Class4 c4);
};

class Class2
{

private:
    int val;

public:
    void setValue(int value)
    {
        val = value;
    }
    friend void max(Class1 c1, Class2 c2, Class3 c3, Class4 c4);
};

class Class3
{

private:
    int val;

public:
    void setValue(int value)
    {
        val = value;
    }
```

```
    friend void max(Class1 c1, Class2 c2, Class3 c3, Class4 c4);
};

class Class4
{

private:
    int val;

public:
    void setValue(int value)
    {
        val = value;
    }
    friend void max(Class1 c1, Class2 c2, Class3 c3, Class4 c4);
};

void max(Class1 c1, Class2 c2, Class3 c3, Class4 c4)
{

    if (c1.val > c2.val && c1.val > c3.val && c1.val > c4.val)
    {
        cout << "Class 1 value is maximum";
    }

    else if (c2.val > c3.val && c2.val > c4.val)
    {

        cout << "Class 2 value is maximum";
    }

    else if (c3.val > c4.val)
    {

        cout << "Class 3 value is maximum";
    }

    else
    {
        cout << "Class 4 value is maximum";
    }
}

int main()
{
    Class1 class1;
    Class2 class2;
    Class3 class3;
    Class4 class4;
```

```
    class1.setValue(1);
    class2.setValue(2);
    class3.setValue(3);
    class4.setValue(4);

    max(class1, class2, class3, class4);
}
```

Q9. Create a class called Mountain with data members name, height and location. Define a member function to initialize the data members. A friend function cmpHeight() to compare height of two objects and member function displayInfo() to display information of mountain. In main create two objects of the class mountain and display the information of mountain with greatest height.

```
#include <iostream>

using namespace std;

class Mountain
{

private:
    char name[10], location[15];
    int height;

public:
    void setInfo()
    {

        cout << "Enter Mountain name : ";
        cin >> name;
        cout << "Enter Mountain height : ";
        cin >> height;
        cout << "Enter Mountain Location : ";
        cin >> location;
    }
    void displayInfo()
    {

        cout << endl
             << "Name : " << name << endl
             << " Height : " << height << endl
             << " Location : " << location;
    }
    friend void cmpHeight(Mountain m1, Mountain m2);
```

```
};

void cmpHeight(Mountain m1, Mountain m2)
{
        (m1.height > m2.height) ? m1.displayInfo() : cout << m2.displayInfo();
}

int main()
{
   Mountain m1, m2;

   m1.setInfo();
   m2.setInfo();

   cout << endl
        << "================\n Given Info \n\n";
   m1.displayInfo();
   m2.displayInfo();
   cout << "====================\n";

   cmpHeight(m1, m2);

   return 1;
}
```

Q10. Write a programto count the total number of students that have registered their names to playing football in sports week. Define a class that has static data member to count the number of students and define member functions as required.

```
#include <iostream>

using namespace std;
class Student
{

public:
   char name[20];
   static int playerNo;
   void getInfo()
   {
      cout << "Enter student Name : ";
      cin >> name;
      playerNo++;
   }

   void totalPlayers()
   {
```

```
        cout << "Total players : " << playerNo;
    }
};

int Student ::playerNo;

int main()
{
    Student obj, obj2, obj3;

    obj.getInfo();
    obj2.getInfo();
    obj3.totalPlayers();

    return 1;
}
```