

COM102A/BLM104  
LAB 5  
SAMPLE QUESTIONS/ÖRNEK SORULAR

1) In this question, you are supposed to write “Rational.cpp” and “main.cpp” files to be consistent with “Rational.h” in Figure 1.

Overload the addition operator (+) to add two Rational numbers.

Overload the subtraction operator (-) to subtract two Rational numbers.

Overload the multiplication operator (\*) to multiply two Rational numbers.

Overload the ostream operator (<<), to print Rational Numbers as “nominator/denominator” e.g. 3/5.

```
1 #include<iostream>
2 using namespace std;
3
4 class Rational
5 {
6 private:
7     int numerator,denominator;
8 public:
9     Rational(int ,int);    //constructor
10    void setFraction(int ,int); //set data members
11    Rational operator+(Rational); //overload +, return new object with new values
12    Rational operator-(Rational); //overload -, return new object with new values
13    Rational operator*(Rational); //overload *, return new object with new values
14    void reduction(void);
15
16    friend ostream &operator<<(ostream &,const Rational&); //overloaded <<
17 };|
18
```

Figure1 - Rational.h (Şekil1 - Rational.h)

The example output is given in Figure 2.

```
2 3 1 15
Toplam : 11/15
Fark : 3/5
Carpim : 2/45
```

Figure 2 - Output (Şekil2 - Çıktı)

(Bu soruda, Şekil1’de verilen “Rational.h” sınıfına uygun “Rational.cpp” ve “main.cpp” dosyalarını oluşturmanız beklenmektedir.

+ işlemini iki Rational sayıyı toplayacak şekilde yükleyiniz.

- işlemini iki Rational sayıyı çıkaracak şekilde yükleyiniz.

\* işlemini iki Rational sayıyı çarpacak şekilde yükleyiniz.

<< işlemini Rational sayıları “pay/payda” yazacak şekilde yükleyiniz, örneğin 3/5.

Şekil2’de örnek bir çıktı verilmiştir.)

2) In this question, you are supposed to write “Complex.cpp” and “main.cpp” files to be consistent with “Complex.h” in Figure 3.

Overload the addition operator (+) to add two Complex numbers.

Overload the subtraction operator (-) to subtract two Complex numbers.

Overload the ostream operator (<<), to print Complex Numbers as “real + i imaginary” e.g. 3 + i5.

```

#include<iostream>
using namespace std;

class Complex {
    private:
        double real;
        double imag;

    public:
        Complex(double, double);
        double getReal();
        void setReal(double real);
        double getImag();
        void setImag(double imag);

        friend Complex operator+(const Complex &c1, const Complex &c2);
        friend Complex operator-(const Complex &c1, const Complex &c2);
        friend std::ostream& operator<< (std::ostream &out, const Complex &obj);
};

```

Figure3 - Complex.h (Şekil3 - Complex.h)

The example output is given in Figure 4.

```

Toplam : 12 + i4
Fark : 5 + i9

```

Figure4 - Output (Şekil4 - Çıktı)

(Bu soruda, Şekil3’de verilen “Complex.h” sınıfına uygun “Complex.cpp” ve “main.cpp” dosyalarını oluşturmanız beklenmektedir.

+ işlemini iki Complex sayıyı toplayacak şekilde yükleyiniz.

- işlemini iki Complex sayıyı çıkaracak şekilde yükleyiniz.

<< işlemini Complex sayıları “reel + i sanal” yazacak şekilde yükleyiniz, örneğin 3 + i5. Şekil4’de örnek bir çıktı verilmiştir.)

3) In this question, you are supposed to write “Point.cpp” and “main.cpp” files to be consistent with “Point.h” in Figure 5.

Overload the addition operator (+) to add two Point objects.

Overload the subtraction operator (-) to subtract two Point objects.

Overload the ostream operator (<<), to print Point objects as “P(3,5)” for (3,5).

Overload the paranthesis operator (), to reset Point object as (0,0).

```

#include<iostream>
using namespace std;

class Point
{
    public:
        Point(int, int);
        void set_x(int);
        void set_y(int);
        int get_x();
        int get_y();
        void operator()();

        friend Point operator+(const Point &p1, const Point &p2);
        friend Point operator-(const Point &p1, const Point &p2);
        friend std::ostream& operator<< (std::ostream &out, const Point &point);

    private:
        double x,y;
};

```

Figure5 - Point.h (Şekil5 – Point.h)

The example output is given in Figure 6.

```
3
5
P(3, 5)
Toplam = P(5, 8)
Fark = P(1, 2)
P(0, 0)
```

Figure6 - Output (Şekil6 - Çıktı)

(Bu soruda, Şekil5’de verilen “Point.h” sınıfına uygun “Point.cpp” ve “main.cpp” dosyalarını oluşturmanız beklenmektedir.

+ işlemini iki Point nesnesini toplayacak şekilde yükleyiniz.

- işlemini iki Point nesnesini çıkaracak şekilde yükleyiniz.

<< işlemini herhangi bir Point nesnesini örneğin (3,5) noktası için “P(3,5)” yazacak şekilde yükleyiniz.

() işlemini Point nesnesi değerlerini sıfırlayacak şekilde yükleyiniz.

Şekil6’de örnek bir çıktı verilmiştir.)