

```
#include "Number.h"
#include <iostream>
#include <iomanip>
#include <locale>
#include <algorithm>
#include <sstream>
```

```
Number::Number()
```

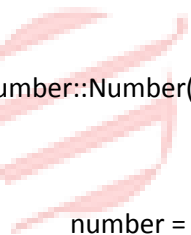
```
{
    number = 0;
    int_to_EURO();
    int_to_US();
}
```

```
Number::Number(int n)
```

```
{
    number = n;
    int_to_US();
    int_to_EURO();
}
```

```
void Number::int_to_US()
```

```
{
    stringstream ss;
    ss.imbue(locale(""));
    ss << fixed << number;
    US = ss.str();
}
```



EssayCorp 5 years ★★★★★

```
void Number::int_to_EURO()
{
    stringstream ss;
    ss.imbue(locale(""));
    ss << fixed << number;
    EURO = ss.str();
    replace(EURO.begin(), EURO.end(), ',', '.');
}
```

```
string Number::get_US_style() const
```

```
{
    return US;
}
```

```
// Returns int as a string in Euro style.
```

```
string Number::get_EURO_style() const
```

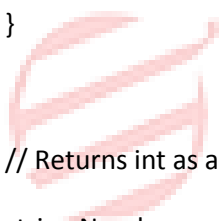
```
{
    return EURO;
}
```

```
// Returns the int.
```

```
int Number::get_number() const
```

```
{
    return number;
}
```

```
// Sets the value of int, US, and EURO.
```



EssayCorp 5 years ★★★★★

```
void Number::set_number(int n)
{
    number = n;
    int_to_US();
    int_to_EURO();
}
```

```
#ifndef NUMBER_H
```

```
#define NUMBER_H
```

```
#include<string>
```

```
using namespace std;
```

```
class Number
```

```
{
```

```
public:
```

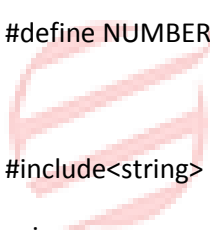
```
    // Default constructor without any initial value.
```

```
    Number();
```

```
    // Constructor with initial value.
```

```
    Number(int);
```

```
    // Returns int as a string in US style.
```



EssayCorp 5 years ★★★★★

```
string get_US_style()const;

// Returns int as a string in Euro style.
string get_EURO_style()const;

// Returns the int.
int get_number()const;

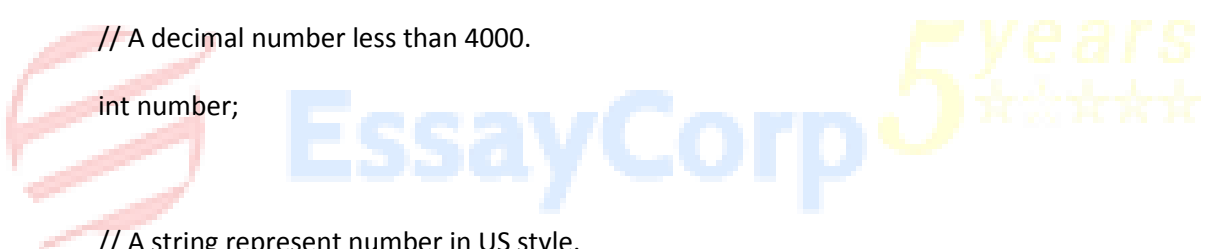
// Sets the value of int, US, and EURO.
void set_number(int);

private:
// A decimal number less than 4000.
int number;
// A string represent number in US style.
string US;

// A string representing number in EURO style.
string EURO;

// converting int to string in US style
void int_to_US();

// converting int to string in EURO style
void int_to_EURO();
};
```



```
#endif
```

```
#include"roman.h"
```

```
Roman::Roman() : Number()
```

```
{
```

```
    roman = "INVALID";
```

```
}
```

```
Roman::Roman(int n) :Number(n)
```

```
{
```

```
    int_to_Roman();
```

```
}
```

```
void Roman::int_to_Roman()
```

```
{
```

```
    int n = get_number();
```

```
    roman = "";
```

```
    if (n <= 0)
```

```
    {
```

```
        roman = "INVALID";
```

```
    }
```

```
    else
```



```

    {
        int th, h, t, o;
        string thousands[] = { "", "M", "MM", "MMM", "MMMM" };
        string hundreds[] = { "", "C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM" };
        string tens[] = { "", "X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC" };
        string ones[] = { "", "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX" };

        th = n / 1000;
        n = n % 1000;

        h = n / 100;
        n = n % 100;

        t = n / 10;
        o = n % 10;

        roman += thousands[th];
        roman += hundreds[h];
        roman += tens[t];
        roman += ones[o];
    }
}

string Roman::get_ROMAN_style()
{
    int_to_Roman();
    return roman;
}

```



EssayCorp 5 years ★★★★★

```
#ifndef ROMAN_H
#define ROMAN_H

#include"Number.h"

class Roman: public Number
{
private:
    string roman;
    void int_to_Roman(); // converting int to string Roman Numeral

public:
    Roman();
    Roman(int n);
    string get_ROMAN_style();
};
#endif

#include"roman.h"
#include<Windows.h>
#include<iostream>

int main()
{
    Roman n;

    cout << "Roman one:\n" << n.get_number() << endl << n.get_US_style() << endl <<
n.get_EURO_style() << endl << n.get_ROMAN_style() << endl;

    n.set_number(2014);
```

```
    cout << "Roman one:\n" << n.get_number() << endl << n.get_US_style() << endl <<
n.get_EURO_style() << endl << n.get_ROMAN_style() << endl;
```

```
    Roman n2(22);
```

```
    cout << "Roman two:\n" << n2.get_number() << endl << n2.get_US_style() << endl <<
n2.get_EURO_style() << endl << n2.get_ROMAN_style() << endl;
```

```
    Roman n3(2402);
```

```
    cout << "Roman three:\n" << n3.get_number() << endl << n3.get_US_style() << endl <<
n3.get_EURO_style() << endl << n3.get_ROMAN_style() << endl;
```

```
    system("pause");
```

```
}
```

```
class number
```

```
{
```

```
protected:
```

```
{
```

```
int a;
```

```
long US,EURO;
```

```
char rom;
```

```
}
```

```
}
```

```
class roman:public number
```

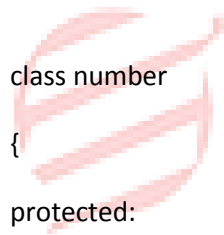
```
{
```

```
public :
```

```
void roman1()
```

```
{
```

```
a=0;
```



EssayCorp **5** years ★★★★★

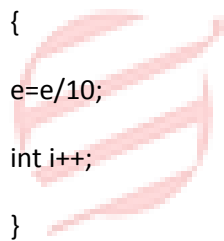

```
cout<<"integer="<<a<<endl;
cout<<"US="<<a<<endl;
cout<<"rom="<<a<<endl;
cout<<"roman="<<"invalid"<<endl;
int b=a+2014;
while(b>0)
{
b=b/10;
int i++;
}
if(i>3)
{
int c=b/1000;
b%=1000;
cout<<"integer="<<b<<endl;
cout<<"US="<<c<<","<<b<<endl;
cout<<"EURO="<<c<<","<<b<<endl;
cout<<"roman="<<"MMXIV"<<endl;
}}
void roman2()
{
int d=a+22;
int j;
while(a>22)
{
d=d/10;
j++;
```



```

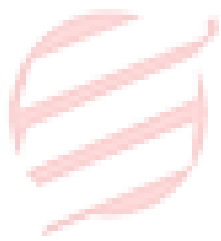
}
if (i<3)
{
cout<<"integer="<<d<<endl;
cout<<"US="<<d<<endl;
cout<<"EURO="<<d<<endl;
cout<<"roman="<<"XXII"<<endl;
}
void roman3()
{
int e=a+2402;
while(b>0)
{
e=e/10;
int i++;
}
if(i>3)
{
int f=e/1000;
e%=1000;
cout<<"integer="<<b<<endl;
cout<<"US="<<f<<","<<e<<endl;
cout<<"EURO="<<f<<","<<e<<endl;
cout<<"roman="<<"MMXIV"<<endl;
}}
}
void main()

```



EssayCorp 5 years ★★★★★

```
{  
roman r;  
r.roman1();  
r.roman2();  
r.roman3();  
}
```



EssayCorp 5 years
★★★★★