LAB ASSIGNMENT - 2

ARZOO KHAN

(20MCA011)

MCA

(Semester - II)

May 7, 2021

—

CSC26: Lab – III (OOP)

—

Dr. S. Zeeshan Hussain

(Associate Professor, D/o Computer Science)

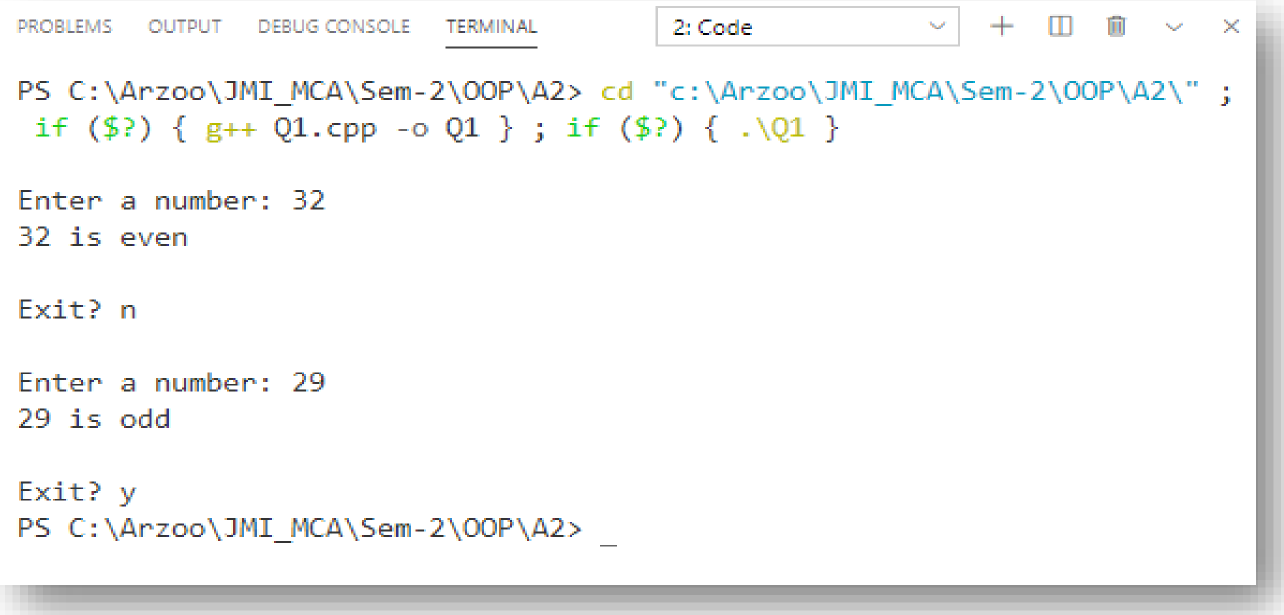
1. ***Check whether a number is even or odd by overloading ‘!’ operator.***

SOURCE CODE:

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

class Integer
{
 private:
    int num;

 public:
    /* Member function to get input from the user for a Integer
    class object */
    void input()
    {
        cout << "\nEnter a number: ";
        cin >> num;
    }

    //Member function to display the Integer Class object
    void display()
    {
        cout << num;
    }


    /* Member function ('!' operator overloading) to check if the
    whether a number is even or not */

    int operator!()
    {
        if (num % 2 == 0)
            return 1;
        else
            return 0;
    }
};
```

```
//Driver Code
int main(void)
{
    Integer x;
    char ex;

    do
    {
        x.input();
        x.display();

        if (!x)
            cout << " is even";
        else
            cout << " is odd";

        cout << "\n\nExit? ";
        cin >> ex;
    } while (ex != 'y');

    return 0;
}
```

OUTPUT:



```
PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A2> cd "c:\Arzoo\JMI_MCA\Sem-2\OOP\A2\" ;
 if ($?) { g++ Q1.cpp -o Q1 } ; if ($?) { .\Q1 }

Enter a number: 32
32 is even

Exit? n

Enter a number: 29
29 is odd

Exit? y
PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A2> _
```

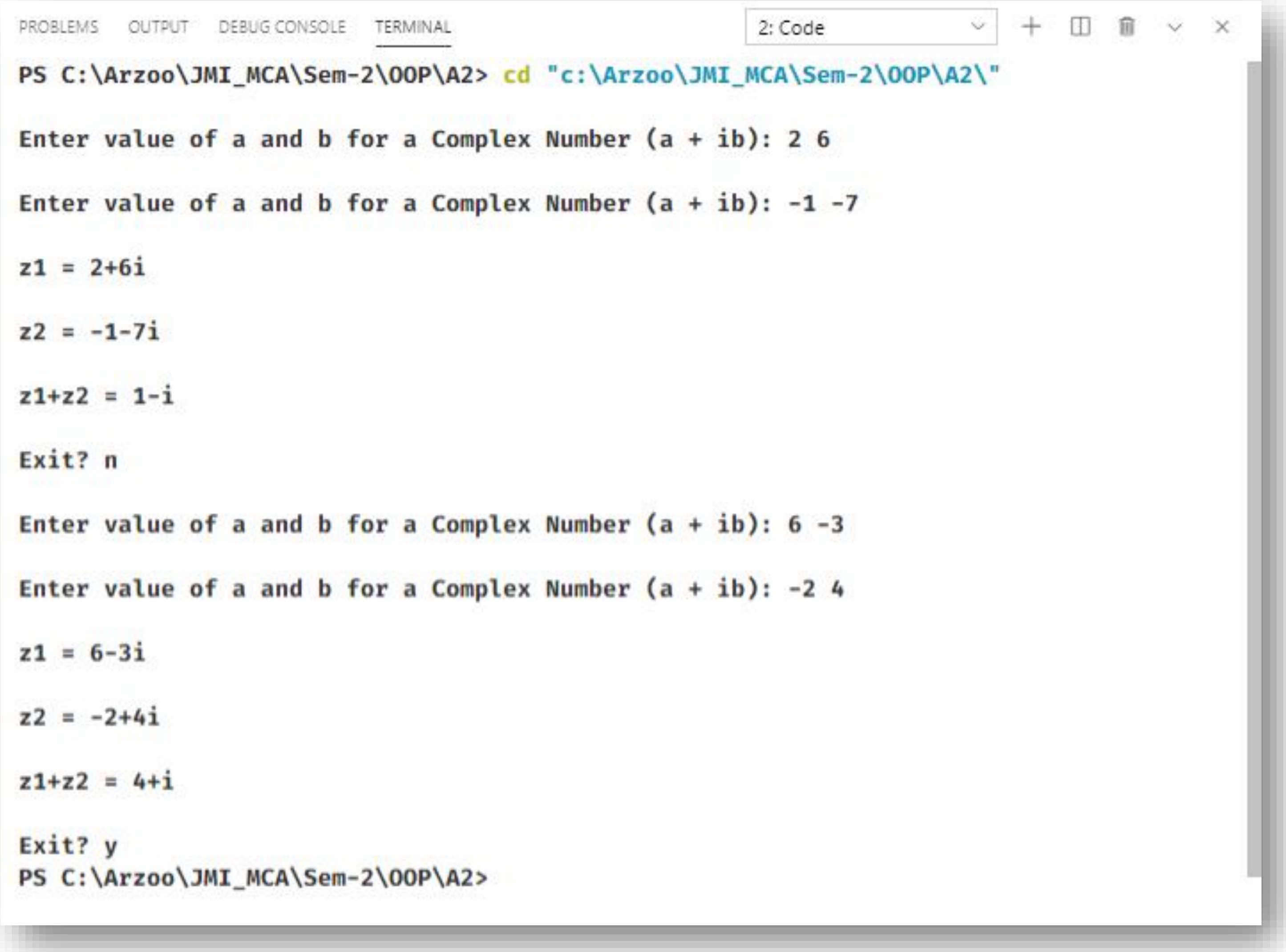
2. ***Write a Program to add two complex number objects of class Complex having real and imaginary as data members by overloading '+' operator***

a) Using Member function.

SOURCE CODE:

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

class Complex
{
private:
    float real, imaginary;

public:
    Complex operator+(Complex &);
    friend istream &operator>>(istream &, Complex &);
    friend ostream &operator<<(ostream &, Complex);
};


// Friend function ('>>' operator overloading) to take input from
//user for Complex class object

istream &operator>>(istream &ccin, Complex &z)
{
    cout << "\nEnter value of a and b for a Complex Number (a + ib): ";
    cin >> z.real >> z.imaginary;
    return (ccin);
}


// Friend function ('<<' operator overloding) to display the
// content of Complex class object

ostream &operator<<(ostream &ccout, Complex z)
{
    if (z.real == 0 && z.imaginary == 0)
        cout << 0;
    else
    {
        if (z.real != 0)
            cout << z.real << (z.imaginary > 0 ? "+" : "");

        if (z.imaginary != 0)
        {
            if (z.imaginary == 1)
                cout << "i";
            else if (z.imaginary == -1)
                cout << "-i";
            else
                cout << z.imaginary << "i";
        }
    }
    return (ccout);
}


// '+' operater overloading using member function

Complex Complex ::operator+(Complex &z)
{
    Complex temp;

    temp.real = real + z.real;
    temp.imaginary = imaginary + z.imaginary;

    return (temp);
}

//Driver Code
int main(void)
{
    Complex z1, z2;
    char ex;
    do
    {
        cin >> z1 >> z2;
        cout << "\nz1 = " << z1 << endl;

        cout << "z2 = " << z2 << endl;

        cout << "z1+z2 = " << z1 + z2 << endl;

        cout << "\n\nExit? ";
        cin >> ex;
    } while (ex != 'y');
    return 0;
}
```

OUTPUT:



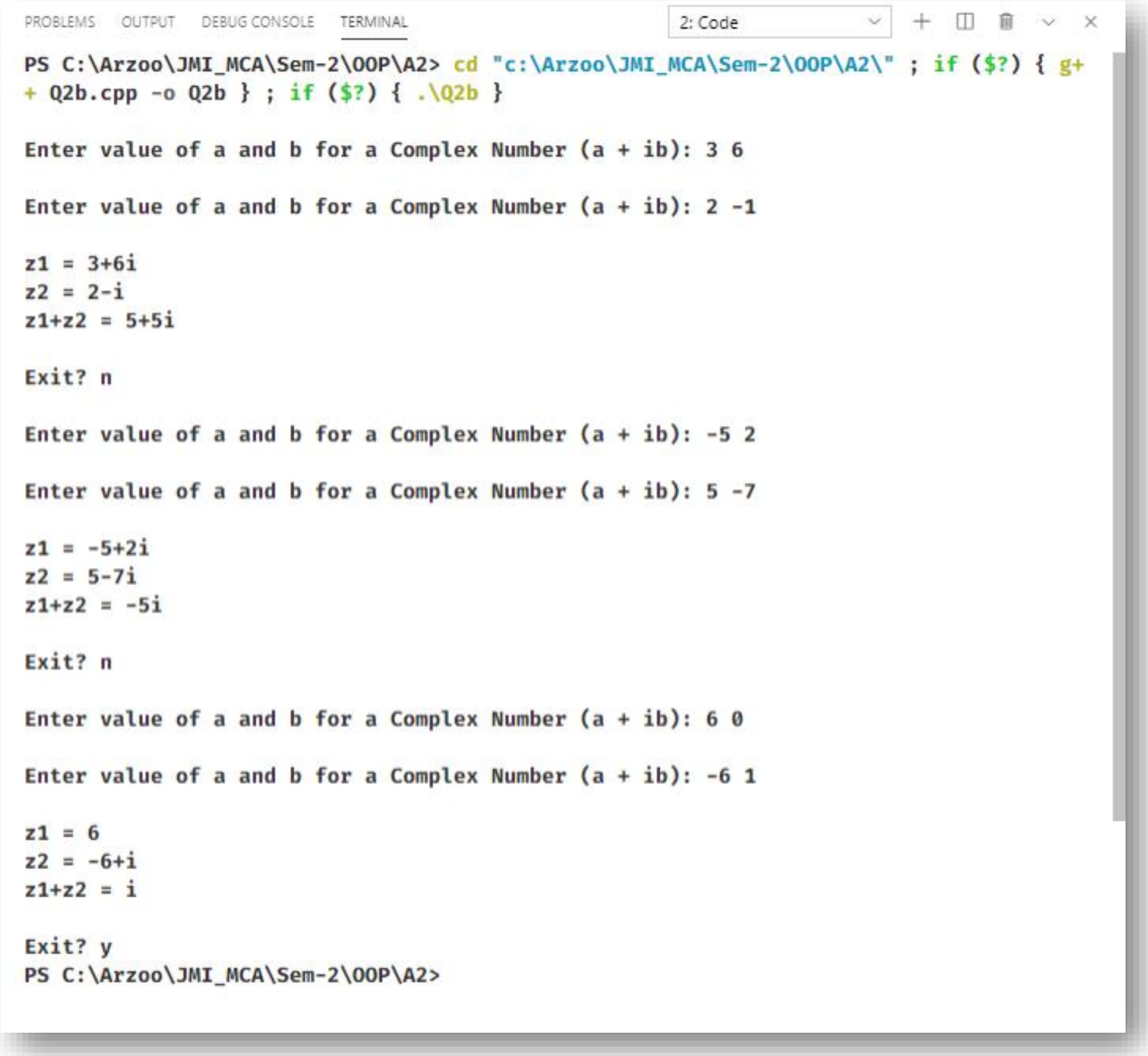
```
PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A2> cd "c:\Arzoo\JMI_MCA\Sem-2\OOP\A2\"

Enter value of a and b for a Complex Number (a + ib): 2 6

Enter value of a and b for a Complex Number (a + ib): -1 -7

z1 = 2+6i

z2 = -1-7i

z1+z2 = 1-i

Exit? n

Enter value of a and b for a Complex Number (a + ib): 6 -3

Enter value of a and b for a Complex Number (a + ib): -2 4

z1 = 6-3i

z2 = -2+4i

z1+z2 = 4+i

Exit? y
PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A2>
```

b) Using Friend Function.

SOURCE CODE:

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

class Complex
{
private:
    int real, imaginary;

public:
    friend Complex operator+(Complex &, Complex &);
    friend istream &operator>>(istream &, Complex &);
    friend ostream &operator<<(ostream &, Complex);
};


// Friend function ('>>' operator overloading) to take input from
// user for Complex class object

istream &operator>>(istream &ccin, Complex &z)
{
    cout << "\nEnter value of a and b for a Complex Number (a + ib): ";
    cin >> z.real >> z.imaginary;
    return (ccin);
}


// Friend function ('<<' operator overloding) to display the
// content of Complex class object

ostream &operator<<(ostream &ccout, Complex z)
{
    if (z.real == 0 && z.imaginary == 0)
        cout << 0;

    else
    {
        if (z.real != 0)
            cout << z.real << (z.imaginary > 0 ? "+" : "");

        if (z.imaginary != 0)
        {
            if (z.imaginary == 1)
                cout << "i";
            else if (z.imaginary == -1)
                cout << "-i";
            else
                cout << z.imaginary << "i";
        }
    }
    return (ccout);
}


// '+' operator overloading using friend function

Complex operator+(Complex &z1, Complex &z2)
{
    Complex temp;

    temp.real = z1.real + z2.real;
    temp.imaginary = z1.imaginary + z2.imaginary;

    return (temp);
}

//Driver Code
int main(void)
{
    Complex z1, z2;
    char ex;
    do
    {
        cin >> z1 >> z2;

        cout << "\nz1 = " << z1 << endl;

        cout << "z2 = " << z2 << endl;
        cout << "z1+z2 = " << z1 + z2 << endl;

        cout << "\n\nExit? ";
        cin >> ex;
    } while (ex != 'y');
    return 0;
}
```

OUTPUT:



```
PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A2> cd "c:\Arzoo\JMI_MCA\Sem-2\OOP\A2\" ; if ($?) { g+
+ Q2b.cpp -o Q2b } ; if ($?) { .\Q2b }

Enter value of a and b for a Complex Number (a + ib): 3 6

Enter value of a and b for a Complex Number (a + ib): 2 -1

z1 = 3+6i
z2 = 2-i
z1+z2 = 5+5i

Exit? n

Enter value of a and b for a Complex Number (a + ib): -5 2

Enter value of a and b for a Complex Number (a + ib): 5 -7

z1 = -5+2i
z2 = 5-7i
z1+z2 = -5i

Exit? n

Enter value of a and b for a Complex Number (a + ib): 6 0

Enter value of a and b for a Complex Number (a + ib): -6 1

z1 = 6
z2 = -6+i
z1+z2 = i

Exit? y
PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A2>
```

3. ***Class Distance consists of length in feet and inches. Class Distance contains***
    - ***one default constructor***
    - ***one parameterized constructor***
    - ***function getdata() to take the value of feet and inches.***
    - ***function show() to display.***

    a) Overload ‘<’ operator to compare the two given distances.

    b) Overload ‘+=’ operator in the Distance class.

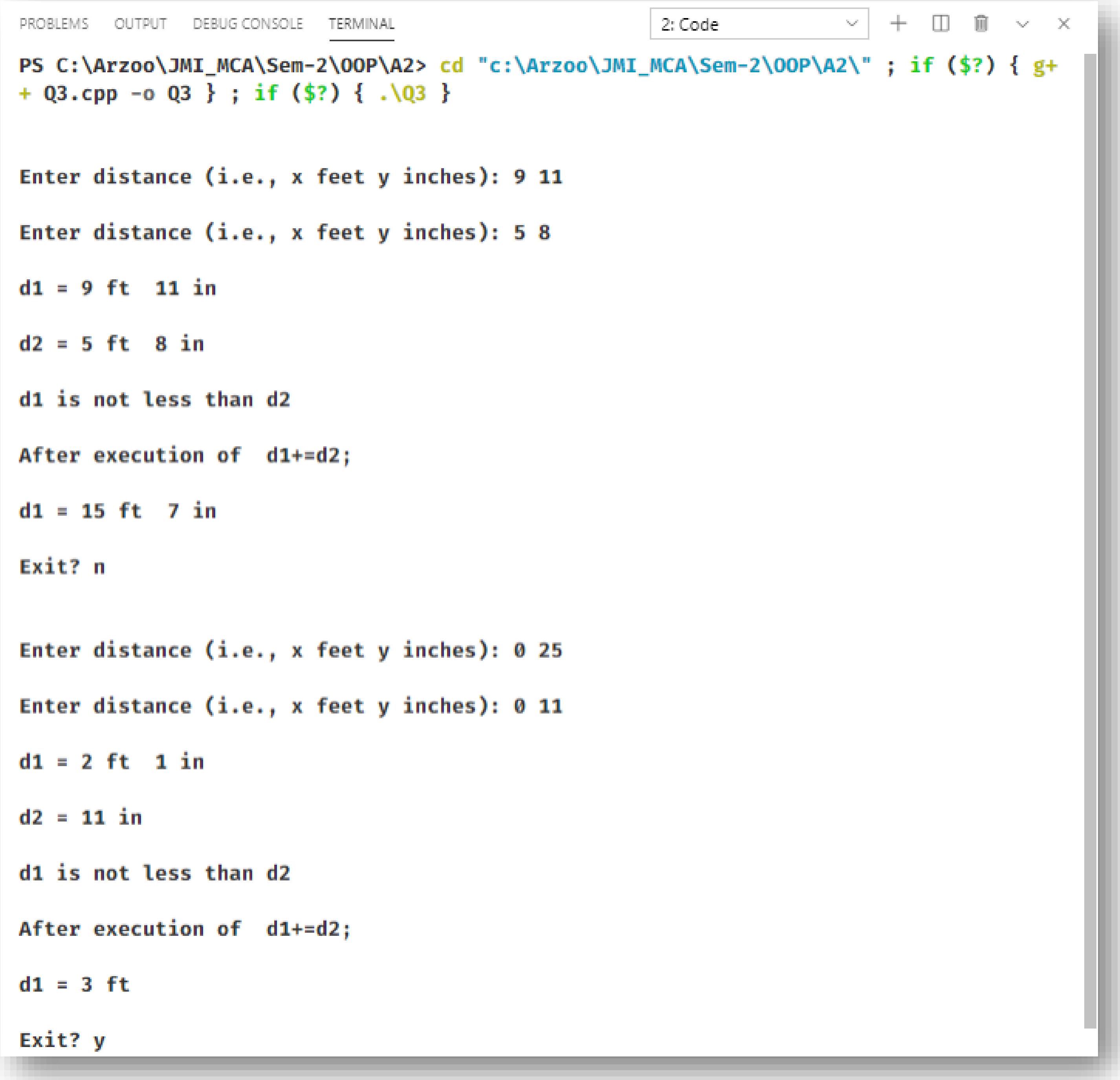
SOURCE CODE:

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

class Distance
{
  private:
      int foot, inch;

  public:
      Distance();
      Distance(int, int);
      void getdata();
      void show();
      friend int operator<(Distance &, Distance &);
      friend void operator+=(Distance &, Distance &);
};


// Default Constructor
Distance ::Distance()
{
}


// Parameterized Constructor
Distance ::Distance(int ft, int in)
{
    foot = ft;
    inch = in;
}
```

```
// Member function to get data from the user

void Distance ::getdata()
{
    cout << "\nEnter distance (i.e., x feet y inches): ";
    cin >> foot >> inch;
    if (inch > 12)
    {
        foot += inch / 12;
        inch %= 12;
    }
}


// Member function to show the content of the object of Distance Class

void Distance ::show()
{
    cout << foot << " ft  " << inch << " in\n";
}


// '<' operator overloading to compare two Distance Class objects

int operator<(Distance &d1, Distance &d2)
{
    if (d1.foot < d2.foot)
        return 1;

    else if (d1.foot == d2.foot)
    {
        if (d1.inch < d2.inch)
            return 1;
        else
            return 0;
    }

    else
        return 0;
}
```

```
// '+=' operator overloading

void operator+=(Distance &s1, Distance &s2)
{
    s1.foot += s2.foot;
    s1.inch += s2.inch;

    if (s1.inch > 12)
    {
        s1.foot++;
        s1.inch %= 12;
    }
}


//Driver Code

int main(void)
{
    Distance d1, d2;
    char ex;
    do
    {
        d1.getdata();
        d2.getdata();

        cout << "\nd1 = ";
        d1.show();
        cout << "\nd2 = ";
        d2.show();

        if (d1 < d2)
            cout << "\nd1 is less than d2\n";
        else
            cout << "\nd1 is not less than d2\n";

        d1 += d2;
        cout << "\nAfter execution of  "
             << "d1+=d2; \n\n"
             << "d1 = ";
        d1.show();

        cout << "\n\nExit? ";
        cin >> ex;
    } while (ex != 'y');
    return 0;
}
```

OUTPUT:



```
PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A2> cd "c:\Arzoo\JMI_MCA\Sem-2\OOP\A2\" ; if ($?) { g+
+ Q3.cpp -o Q3 } ; if ($?) { .\Q3 }


Enter distance (i.e., x feet y inches): 9 11

Enter distance (i.e., x feet y inches): 5 8

d1 = 9 ft  11 in

d2 = 5 ft  8 in

d1 is not less than d2

After execution of  d1+=d2;

d1 = 15 ft  7 in

Exit? n


Enter distance (i.e., x feet y inches): 0 25

Enter distance (i.e., x feet y inches): 0 11

d1 = 2 ft  1 in

d2 = 11 in

d1 is not less than d2

After execution of  d1+=d2;

d1 = 3 ft

Exit? y
```

4. ***Concatenate two string objects of a class String having char str[] and len as data members by overloading '+' operator. Also overload '==', '>' and '<' operators to compare two given String objects.***

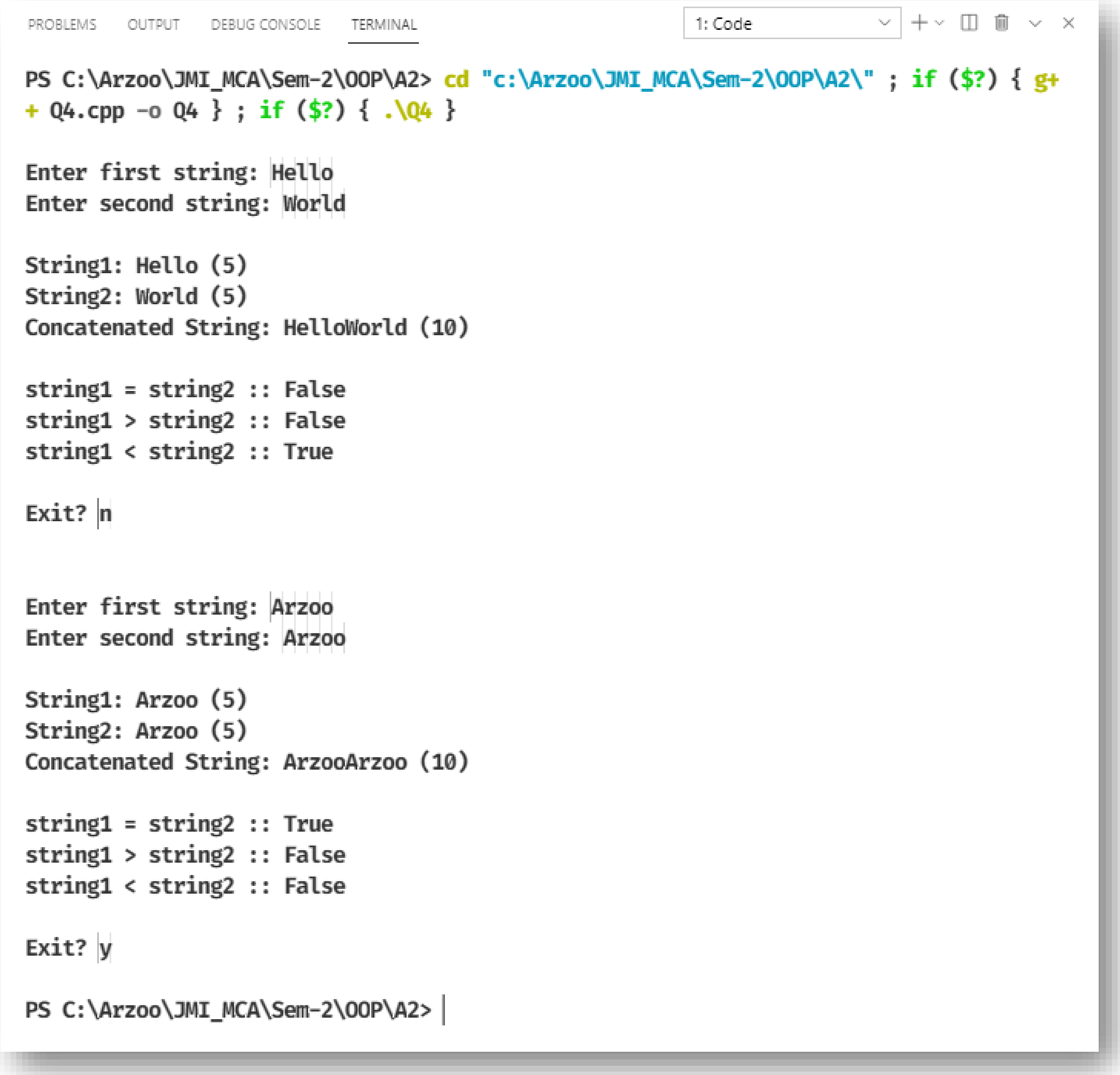
SOURCE CODE:

```
#include <iostream>
#include <conio.h>
#include <string.h>
using namespace std;
#define MAX 20

class String
{
    char str[MAX];
    int len;

public:
    String();
    String(char[]);
    friend istream &operator>>(istream &, String &);
    friend ostream &operator<<(ostream &, String);
    friend String operator+(String, String);
    friend int operator==(String &, String &);
    friend int operator>(String &, String &);
    friend int operator<(String &, String &);
};

// Default Constructor
String ::String()
{
}

// Parameterized Constructor
String ::String(char s[])
{
    strcpy(str, s);
    len = strlen(str);
}
```

```cpp
// '>>' operator overloading to take input from user for a String class object
istream &operator>>(istream &acin, String &s)
{
    cin.getline(s.str, MAX);
    s.len = strlen(s.str);
    return (acin);
}

// '<<' operator overloading to display the object of String class
ostream &operator<<(ostream &acout, String s)
{
    cout << s.str << " (" << s.len << ")";
    return (acout);
}

// Function to concatenate two Strings by '+' operator overloading
String operator+(String a, String b)
{
    strcat(a.str, b.str);
    a.len = strlen(a.str);
    return (a);
}

// Function to compare two Strings by '==' operator overloading
int operator==(String &a, String &b)
{
    return (strcmp(a.str, b.str) == 0 ? 1 : 0);
}

// Function to compare two Strings by '>' operator overloading
int operator>(String &a, String &b)
{
    return (strcmp(a.str, b.str) > 0 ? 1 : 0);
}

// Function to compare two Strings by '<' operator overloading
int operator<(String &a, String &b)
{
    return (strcmp(a.str, b.str) < 0 ? 1 : 0);
}
```

```
//Driver Code
int main(void)
{
    String str1, str2;
    char ex;

    do
    {
        cout << "\nEnter first string: ";
        cin >> str1;
        cout << "Enter second string: ";
        cin >> str2;
        cout << "\nString1: " << str1
             << "\nString2: " << str2
             << "\nConcatenated String: " << str1 + str2 << endl;

        if (str1 == str2)
            cout << "\nstring1 = string2 :: True";
        else
            cout << "\nstring1 = string2 :: False";

        if (str1 > str2)
            cout << "\nstring1 > string2 :: True";
        else
            cout << "\nstring1 > string2 :: False";

        if (str1 < str2)
            cout << "\nstring1 < string2 :: True";
        else
            cout << "\nstring1 < string2 :: False";

        cout << "\n\nExit? ";
        ex = getch();
        cout << ex << "\n\n";
    } while (ex != 'y');

    return 0;
}
```

OUTPUT:



```
PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A2> cd "c:\Arzoo\JMI_MCA\Sem-2\OOP\A2\" ; if ($?) { g++ Q4.cpp -o Q4 } ; if ($?) { .\Q4 }

Enter first string: Hello
Enter second string: World

String1: Hello (5)
String2: World (5)
Concatenated String: HelloWorld (10)

string1 = string2 :: False
string1 > string2 :: False
string1 < string2 :: True

Exit? n


Enter first string: Arzoo
Enter second string: Arzoo

String1: Arzoo (5)
String2: Arzoo (5)
Concatenated String: ArzooArzoo (10)

string1 = string2 :: True
string1 > string2 :: False
string1 < string2 :: False

Exit? y

PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A2> 
```

*****