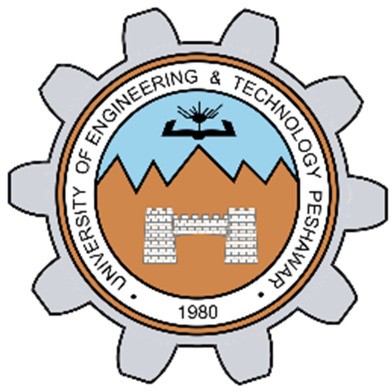
## Object Programming Essentials [Part 2]

## LAB # 02



**Fall 2020**

**CSE208L Object Oriented Programming Lab**

Submitted by: **Fawad Ali** Registration No. : **19PWCSE1845** Class Section: **C**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature:

Submitted to:

## Engr. Sumayyea Salahuddin

December 03, 2020

Department of Computer Systems Engineering University of Engineering and Technology, Peshawar

# Object Programming Essentials [Part 2]

# Objectives of the Lab:

* Understand how class object can be passed and returned from class member function
* Write a class with member function having objects as arguments
* Write a class with member function that return object
* Test member function effectively using given test cases

**Activity # 01**

**Title:**

Create a class and they perform arithmetic operation with complex number. Perform complex addition, subtraction, multiplication.

# Problem analysis:

Create a class, complex that contains a two double real and imaginary. Define a constructor that takes no parameters.

a) Input: Write class function input() to take complex number real and imaginary parts from user on runtime. Note: input takes no arguments and returns nothing.

b) Subtract two Complex numbers: Write class function subCom() taking two complex objects c1 and c2. Difference is computed as following: the real part of the right operand is subtracted from the real part of the left operand, and the imaginary part of the right operand is subtracted from the imaginary part of the left operand. Note: return type of subCom is void.

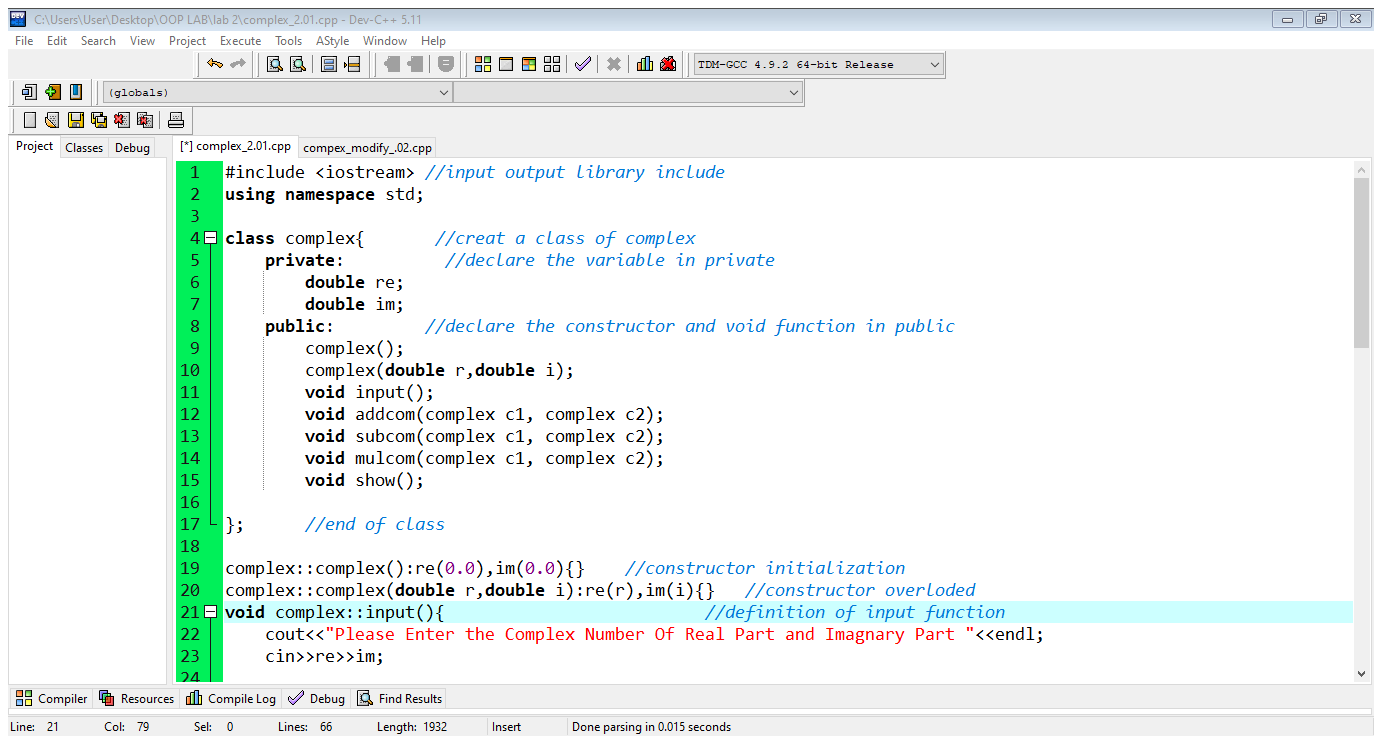
c) Multiply two Complex numbers: Write class function mulCom() taking two complex objects c1 and c2. Product is computed as following: Suppose you are trying to compute the product of two complex numbers a + bi and c + di. The real part will be ac – bd, while the imaginary part will be ad+ bc. Note: return type of subCom is void.

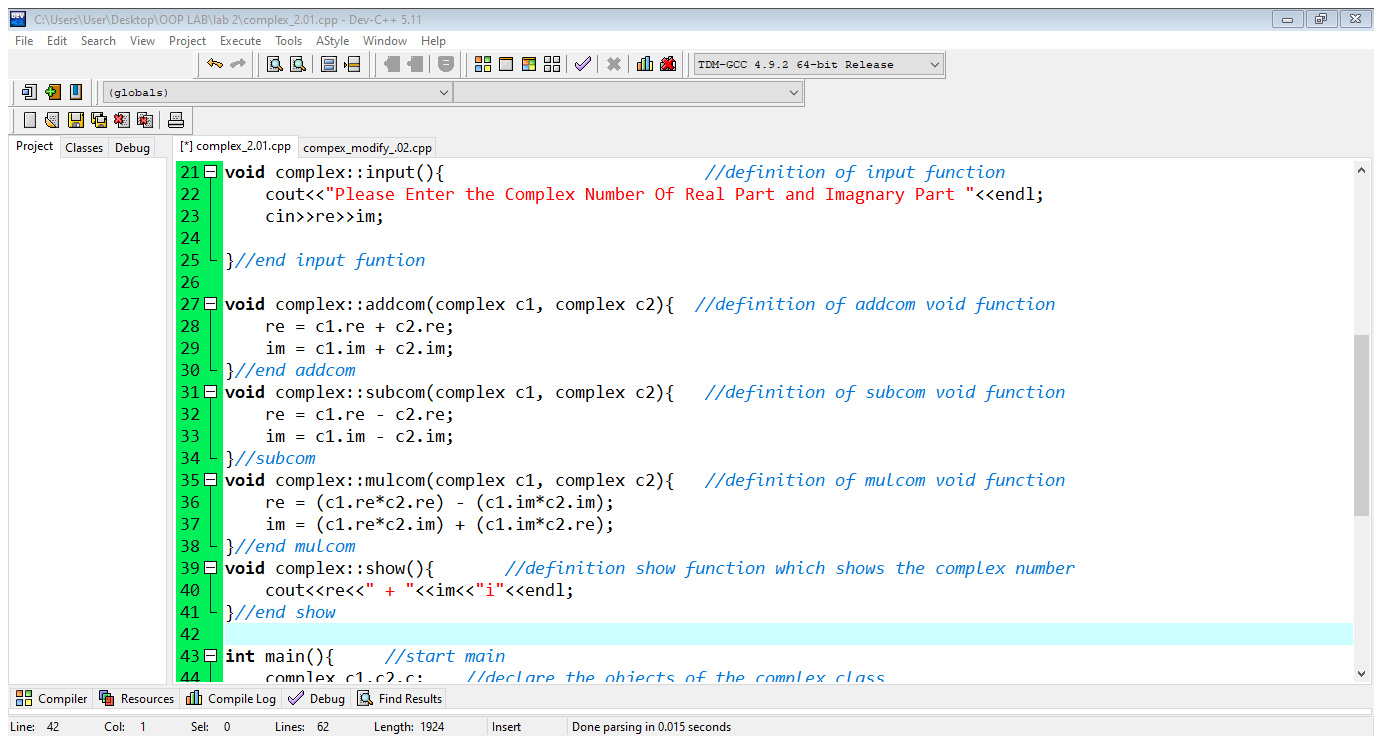
# Algorithm:

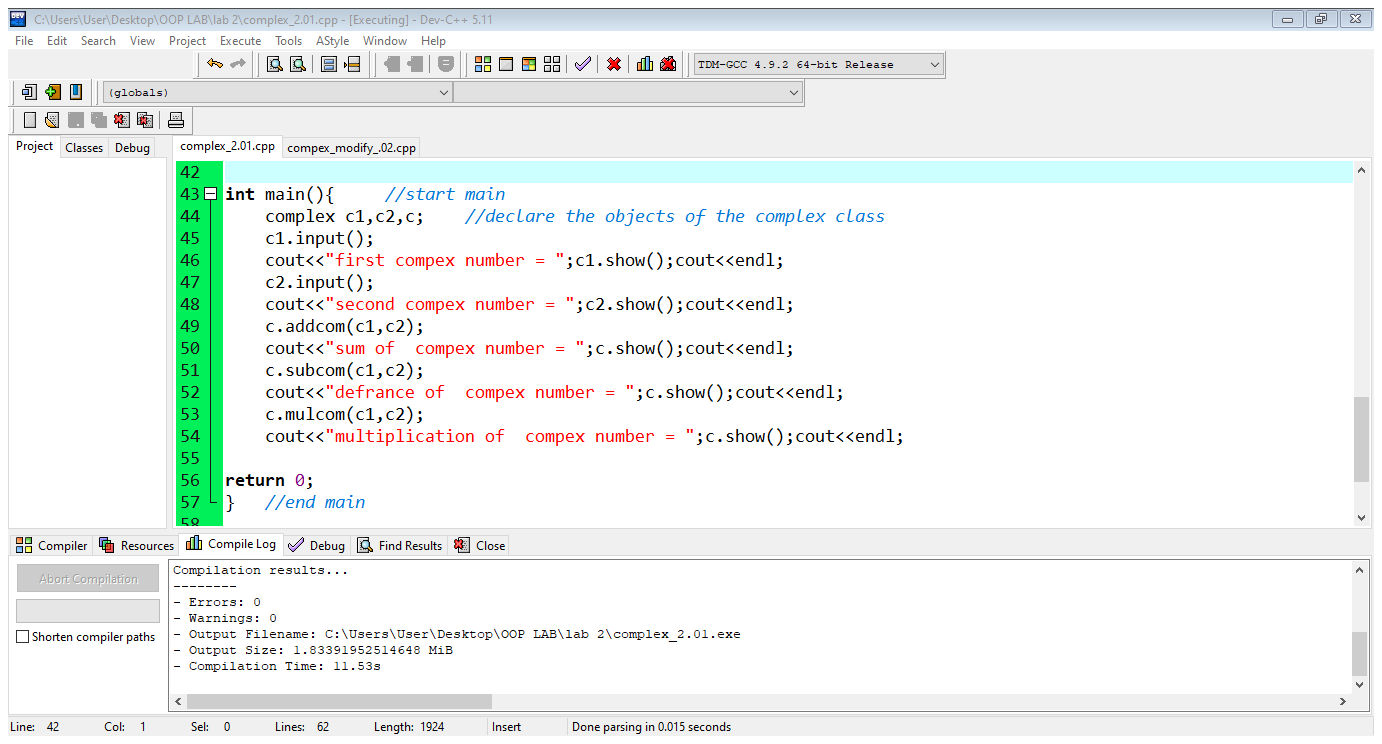
* First make class complex
* Declare the double for real and imaginary part of the complex number.
* Define no argument constructor to set value of real and imaginary is zero.
* Define input function for the input value of complex number.
* Define addcom function for the addition.
* Define subcom function for the subtraction.
* Define mulcom function for the multiplication of complex number.
* Define the show function which display the complex number and addition, subtraction, multiplication of complex number.
* In main function, make objects of the complex c1,c2,c.
* Call each function one after the other and display the show function as shown the complex number.

**In C++**

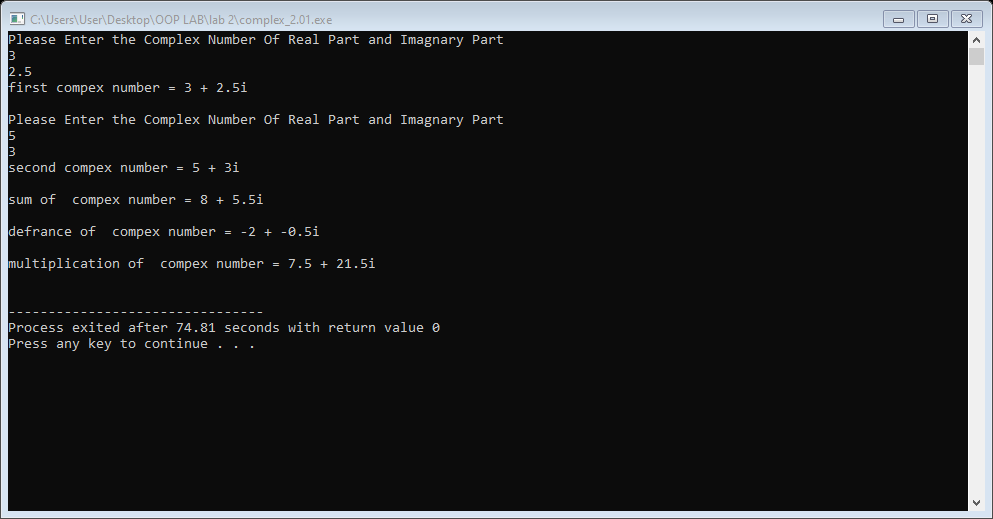
**Source code:**

****

****

****

**Output:**



**Activity # 02**

**Title:**

Reuse Complex class written in Activity 01 to modify the addCom(), subCom(), and mulCom() class functions. Instead of passing two objects in each, now pass only one object. Change the return type of each function to complex. Adjust the function code to match the changes.

# Problem analysis:

Create a class, complex that contains a two double real and imaginary. Define a constructor that takes no parameters.

1. Input: Write class function input() to take complex number real and imaginary parts from user on runtime. Note: input takes no arguments and returns nothing.
2. Addition the two complex number which the one object is pass in they return the object. The real part is add the real part in the imaginary part is add the imaginary.

c) Subtract two Complex numbers: Write class function subCom() taking two complex objects c1, c2 and the another object return . Difference is computed as following: the real part of the right operand is subtracted from the real part of the left operand, and the imaginary part of the right operand is subtracted from the imaginary part of the left operand.

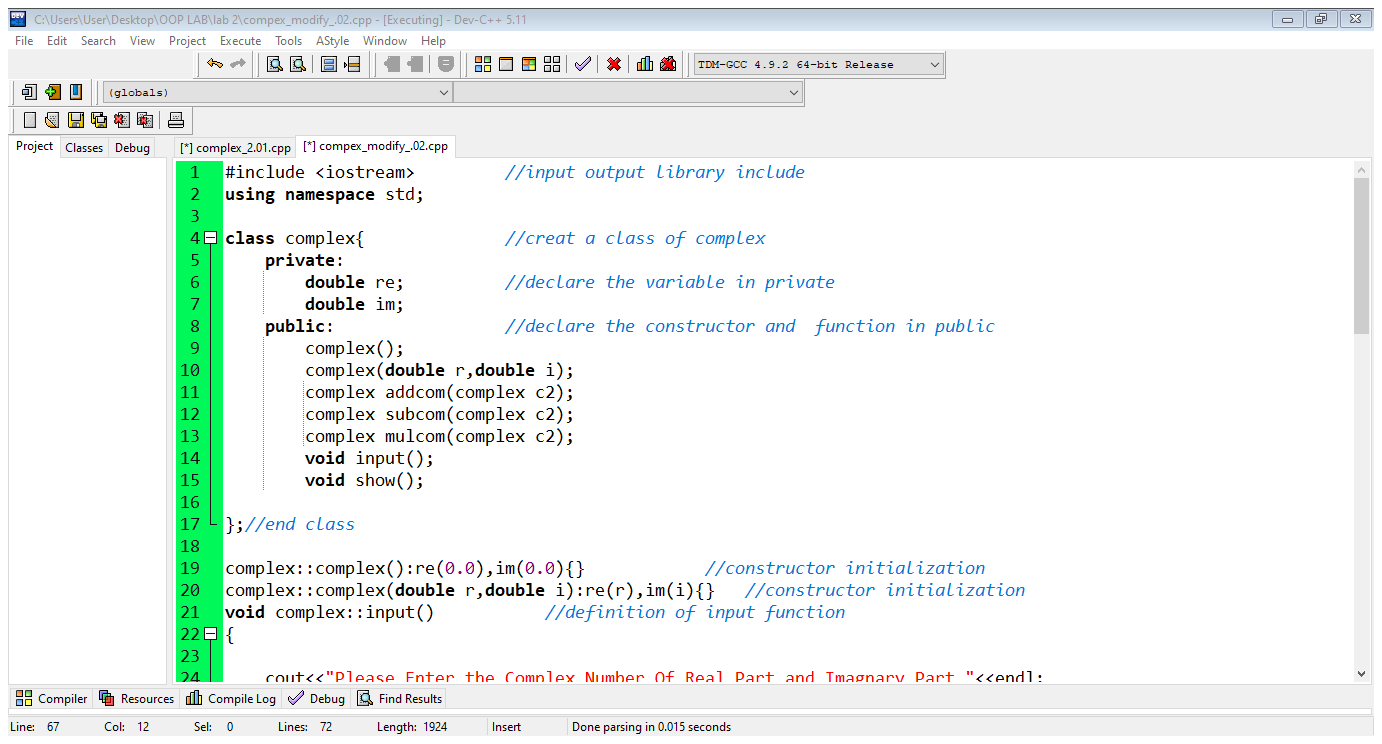
d) Multiply two Complex numbers: Write class function mulCom() taking two complex objects c1 and c2. The one object is pass in the another object bto return . Product is computed as following: Suppose you are trying to compute the product of two complex numbers a + bi and c + di. The real part will be ac – bd, while the imaginary part will be ad+ bc.

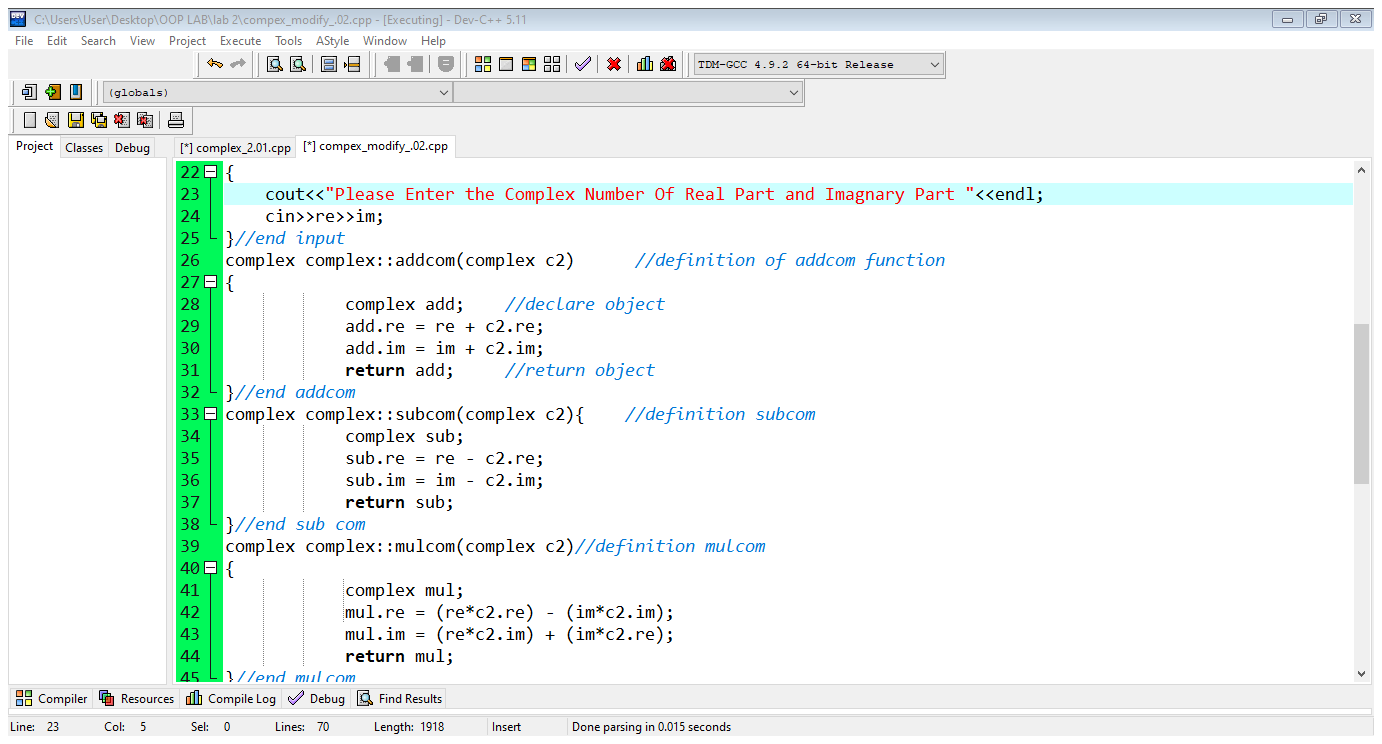
# Algorithm:

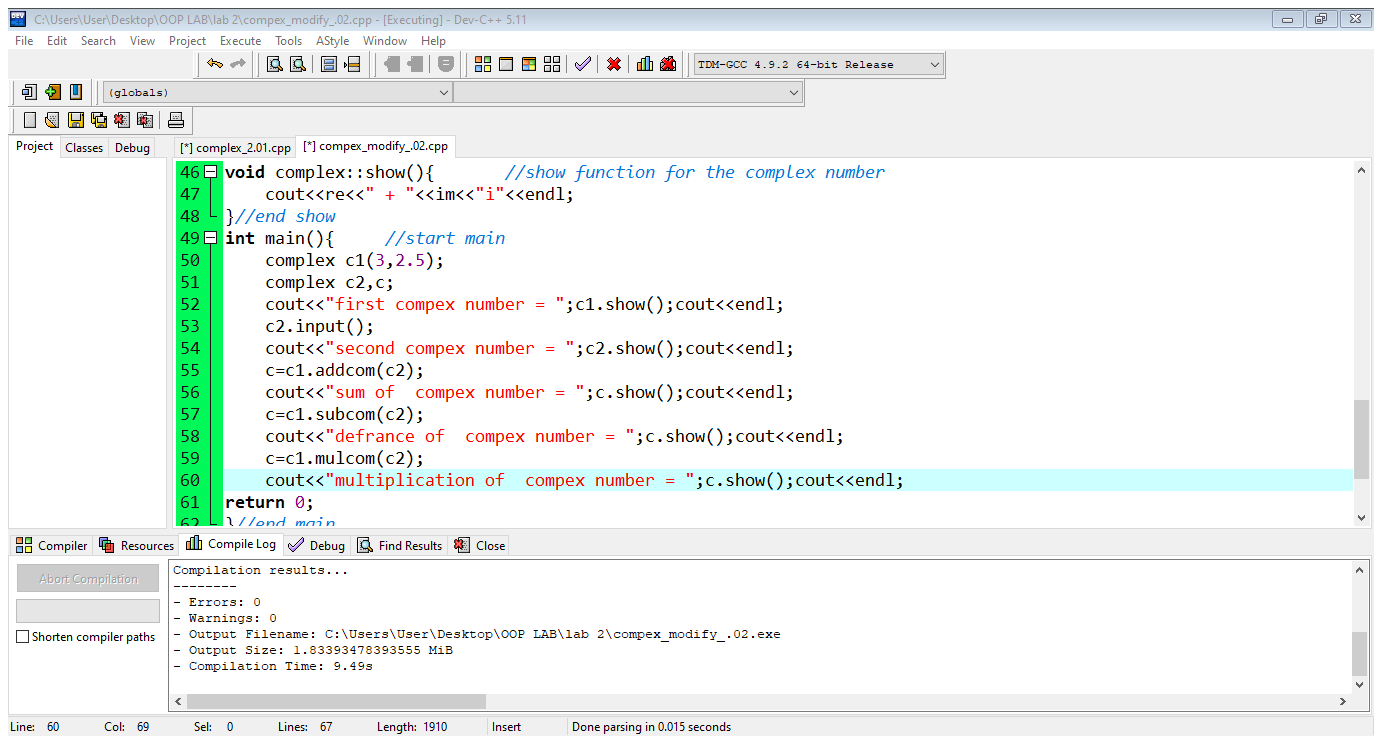
* First make class complex
* Declare the double for real and imaginary part of the complex number.
* Define no argument constructor to set value of real and imaginary is zero.
* Define input function for the input value of complex number.
* Define addcom function for the addition in can return the object.
* Define subcom function for the subtraction in return the object.
* Define mulcom function for the multiplication of complex number and return the object.
* Define the show function which display the complex number and addition, subtraction, multiplication of complex number.
* In main function, make objects of the complex c1,c2,c.
* Call each function one after the other and display the show function as shown the complex number.

**In C++**

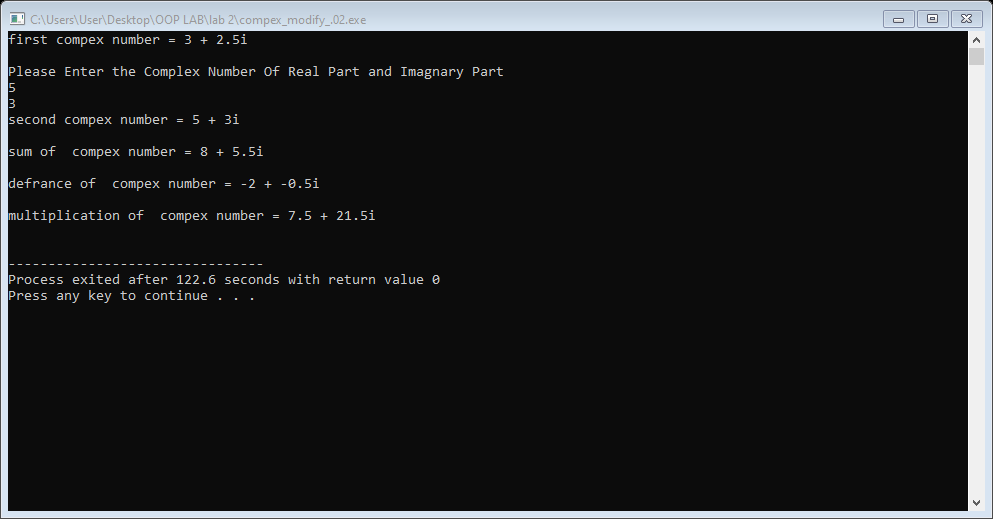
**Source code:**







**Output:**



**Activity # 03**

**Title:**

Create a class called IntegerSet. Each object of class IntegerSet can hold integers in the range 0 through 49. Which can show union,intersection of the integerset.

# Problem analysis:

# Create a class called IntegerSet. Each object of class IntegerSet can hold integers in the range 0 through 49. A set is represented internally as an array of ones and zeros. Array element a[ i ] is 1 if integer i is in the set. Array element a[ j ] is 0 if integer j is not in the set. The default constructor initializes a set to empty set i.e., a set whose array representation contains all zeros.

# Provide member functions for the common set operations. For example,

# 1. Provide a newIntegerSet member function that initialize array to user‐defined array provided as an input in array notation. Note function return type is void.

# 2. Provide a unionOfIntegerSets member function that creates a third set which is the set‐ theoretic union of two existing sets (i.e., an element of the third set’s array is set to 1 if that element is 1 in either or both of the existing sets, and an element of the third set’s array is set to 0 if that element is 0 in each of the existing sets).

# 3. Provide an intersectionOfIntegerSets member function that creates a third set which is the set‐ theoretic intersection of two existing sets (i.e., an element of the third set’s array is set to 0 if that element is 0 in either or both of the existing sets, and an element of the third set’s array is set to 1 if that element is 1 in each of the existing sets).

# 4. Provide an insertElement member function that inserts a new integer k into a set (by setting a[k] to 1).

# 5. Provide a deleteElement member function that deletes integer m (by setting a[m] to 0).

# 6. Provide a setPrint member function that prints a set as a list of numbers separated by spaces. Print only those elements that are present in the set (i.e., their position in the array has a value of 1).

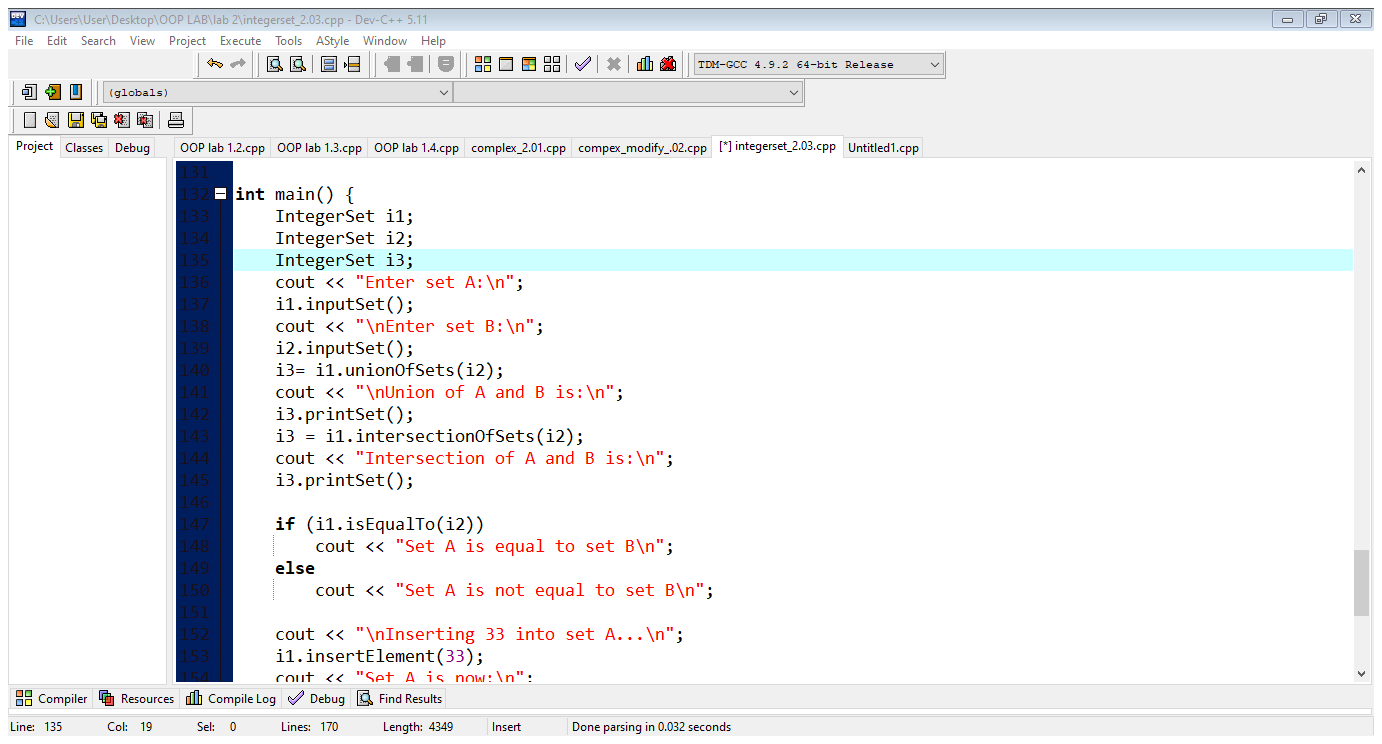
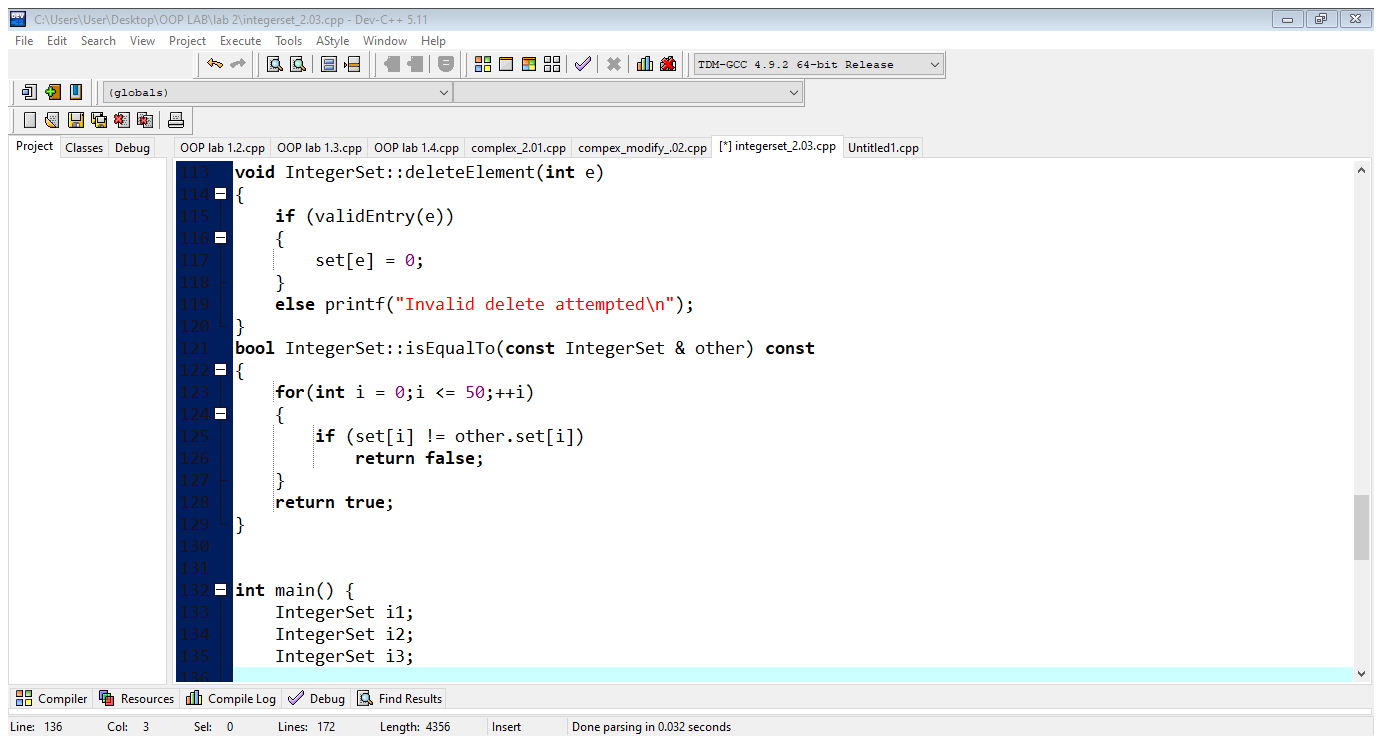
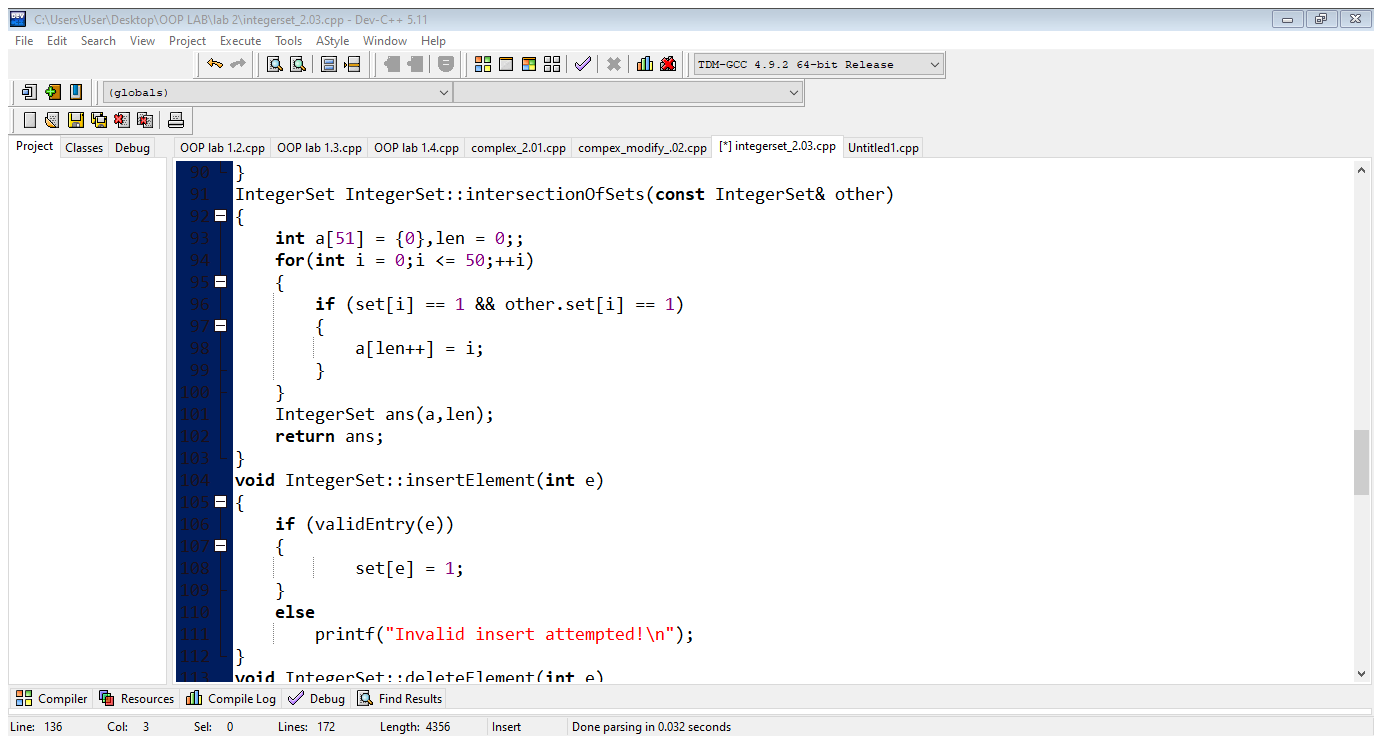
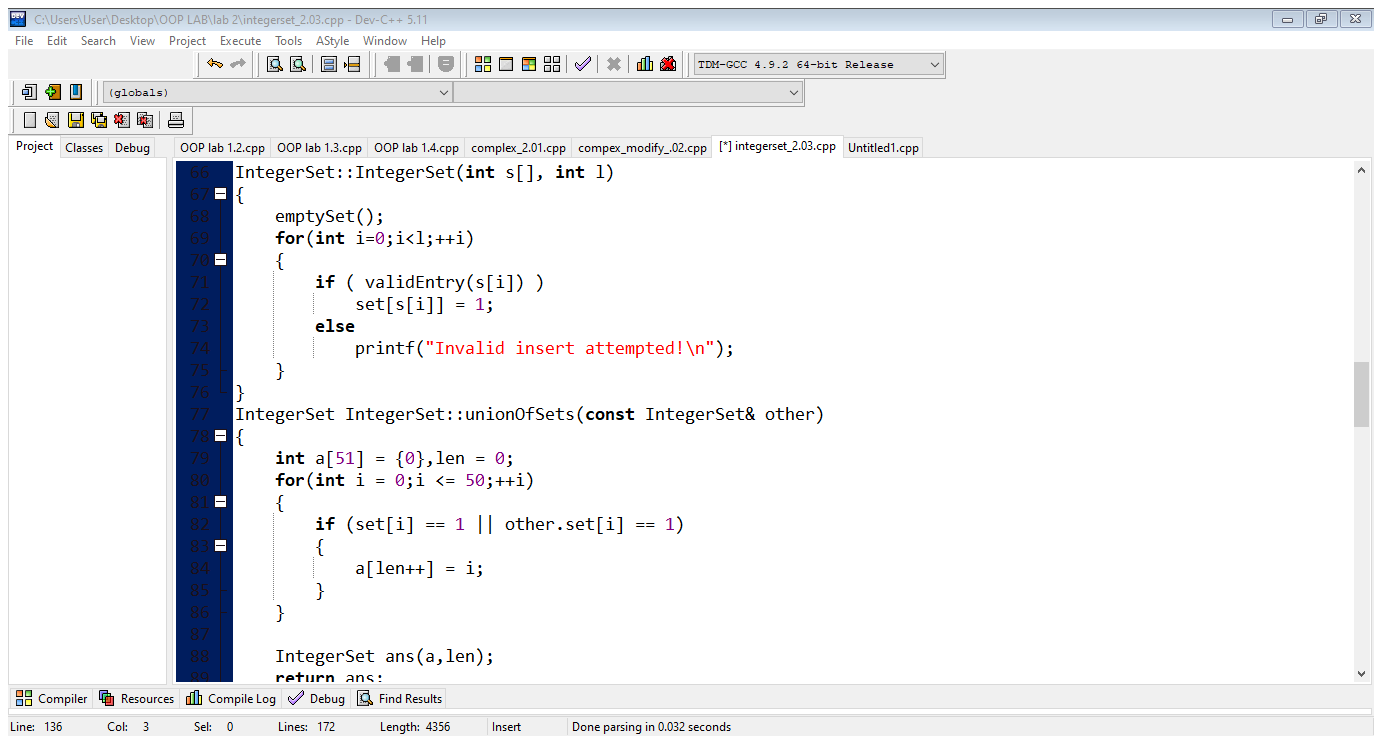
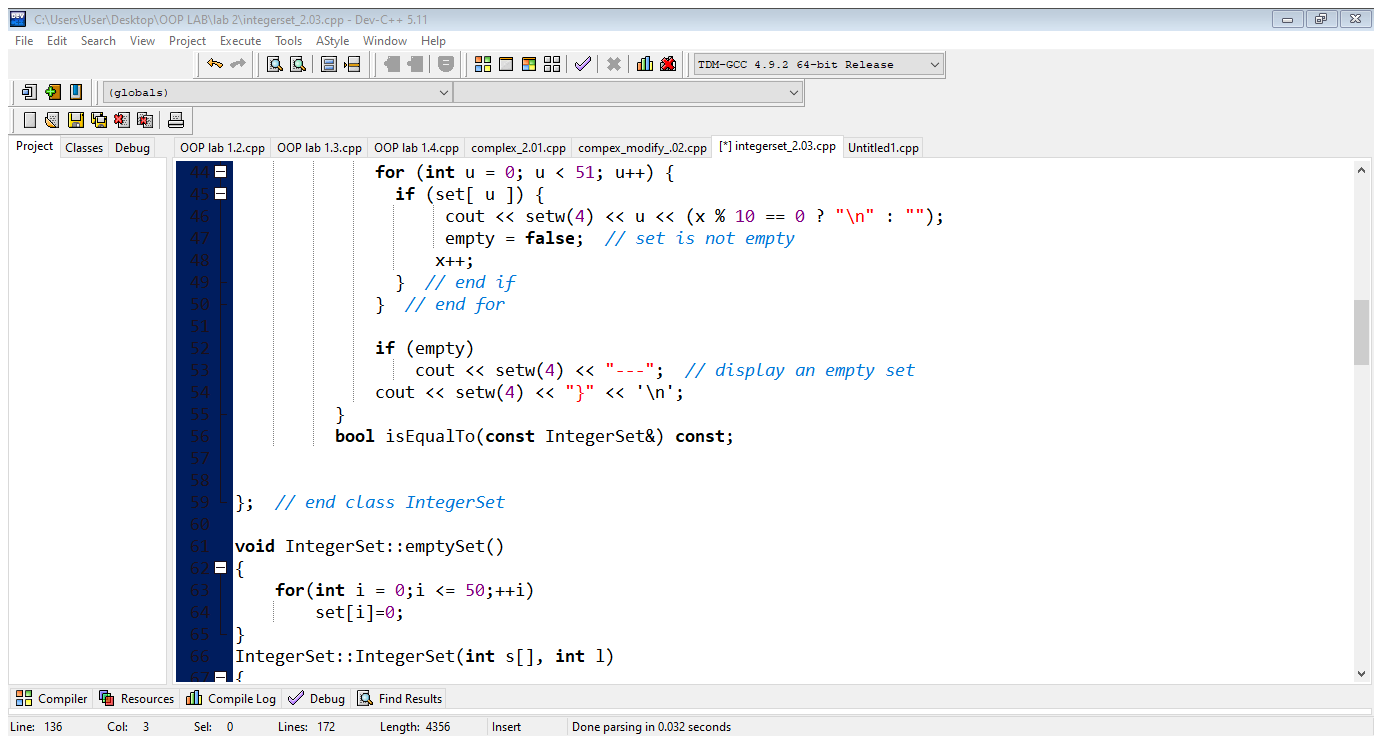
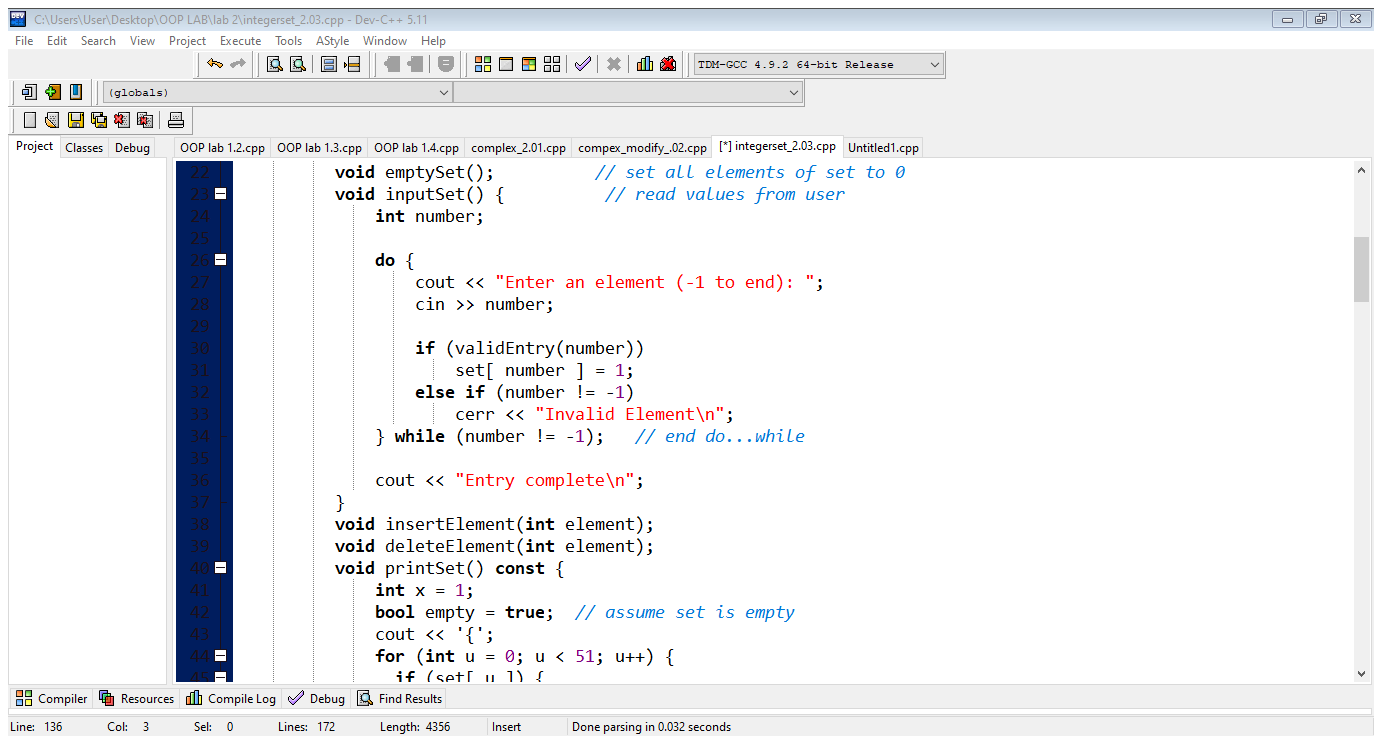
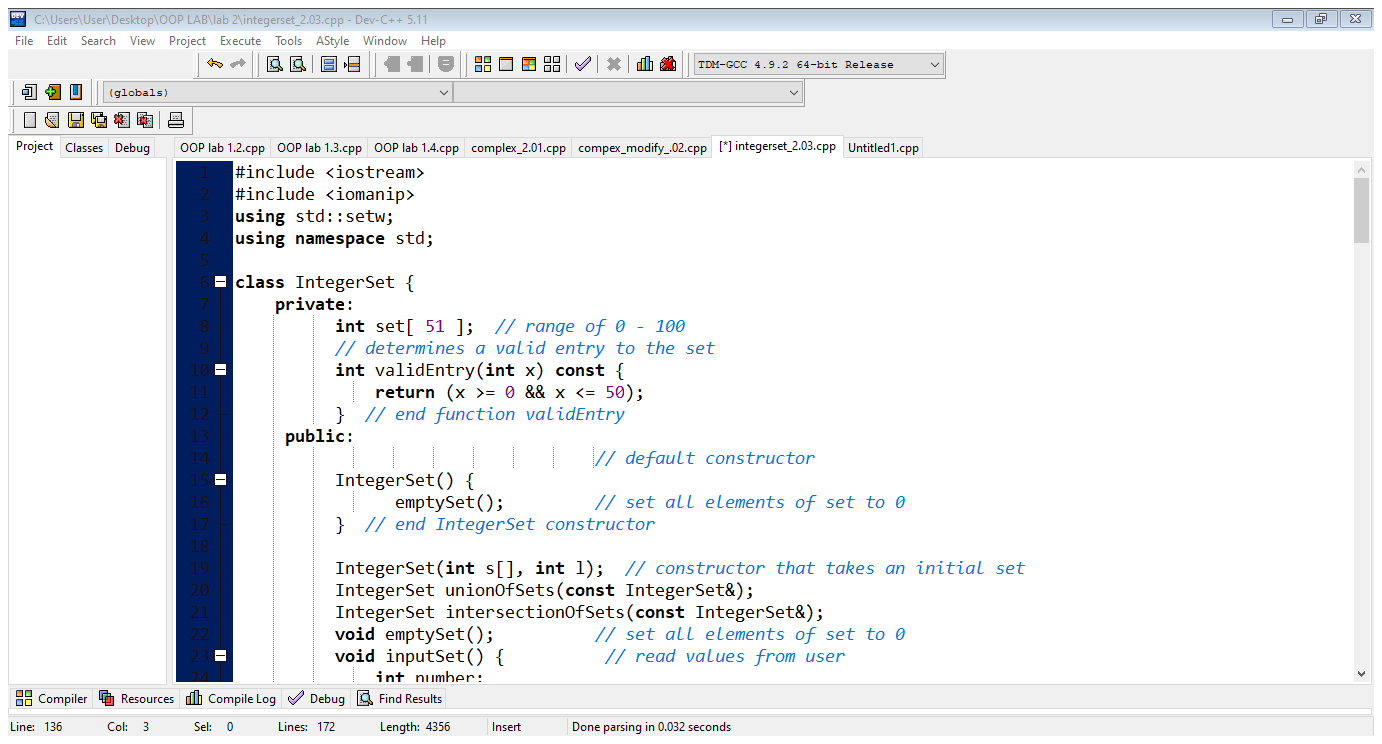
# 7. Provide an isEqualTo member function that determines if two sets are equal.

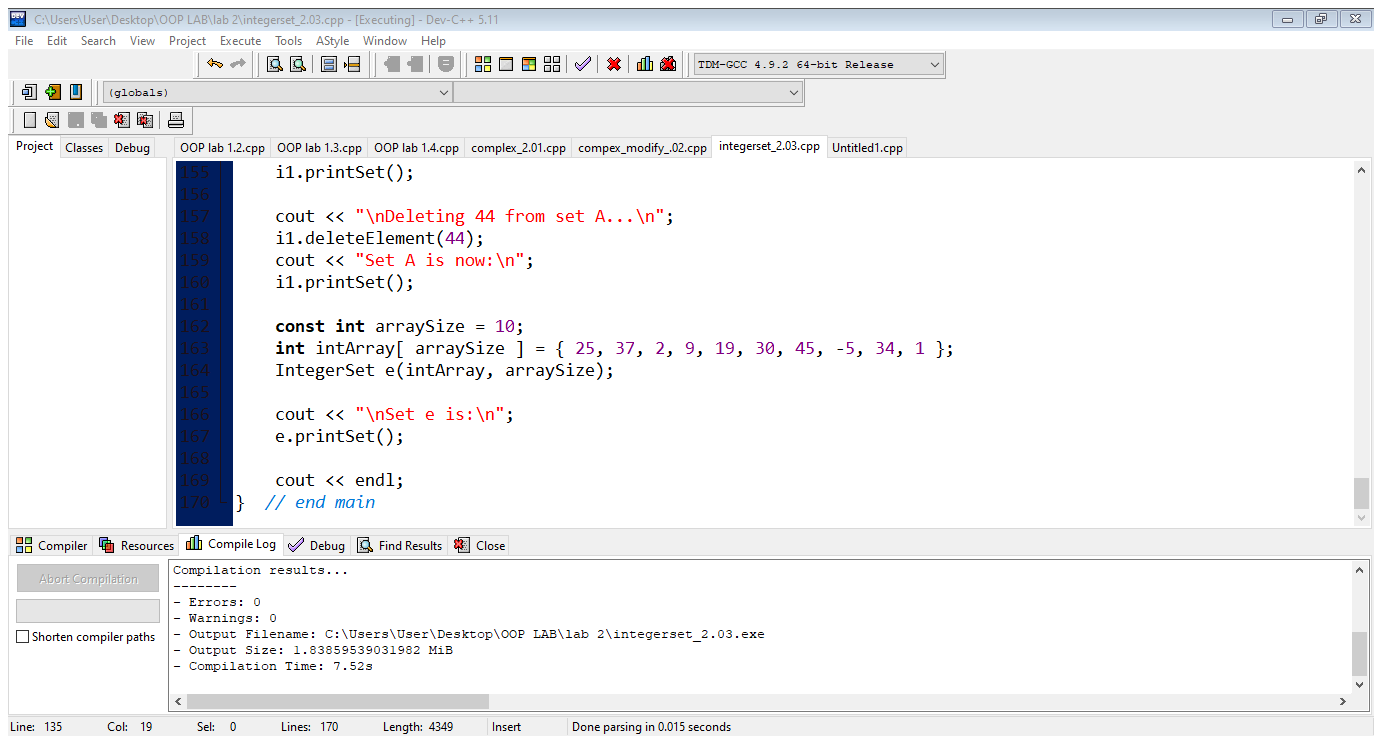
# Algorithm:

* First make class integerset.
* Declare the int set array which range 0 to 49.
* Define constructor to initialize to set of empty set. a[i]=0
* Define input function for the input value of integer number.
* Define integerset function for the value to set the integer. a[i]=1
* Define union function for the to find the union of set a and set b.
* Define intersection function for the intersection of integer set a and b.
* Define the insert function to insert the value which zero is equal to one.
* Define the delete function to the value which one is equal to zero.
* Define equal function to the set a and b equal to show equal are not equal to show not equal.
* Define the print function which display the integer set of a, b, union, intersection all set which call one by one.
* In main function, make objects of the integerset i1,i2,i3.
* Call each function one after the other and display the integer sets.

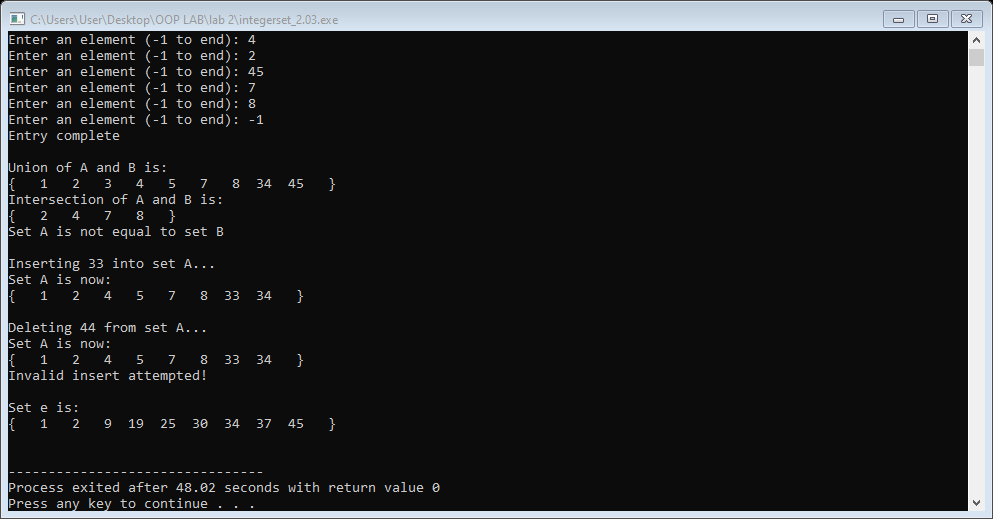
**In C++**

**Source code:**

****

****

**Output:**

****

**Conclusion:**

This program helps us in understanding the basic concepts of classes and objects. when know the constructor and the abject pass and return object and can declare the function and can define out side the class. We check the above activity to know about the all.

**Registration #:**

**Name & Section:**

**Date:**

## CSE 208L – OBJECT ORIENTED PROGRAMMING LAB LAB 01 ASSESSMENT RUBRICS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dimension** | **Exemplary** | **Acceptable** | **Developing** | **Unsatisfactory** | **Student Score out of 10 Marks** |
| **10** | **9‐7** | **6‐5** | **4‐0** |
| **Submission** | Report is submitted on time and in correct format. | Report is submitted on time with slight incorrect format. | Report is submitted on time in incorrect format. | Report is not submitted. |  |
| **Overall Impression of Lab Report** | Report is complete, well written, and organized appropriately with additional elements that enhance it. Task titles and output screenshots are included. Purpose for each concept, input requirements and output results is noted. | Report is complete, briefly written, and organized. Lacks additional elements. Task titles and output screenshots are included. Purpose for each concept, input requirements and output results is noted. | Report is mostly complete, loosely written, and fairly organized. Basic documentation including descriptions of all concepts. Specific purpose is noted for each concept. Task titles and output screenshots are  included and good formatting. | Report is incomplete, sloppy, and/or disorganized.  No documentation included.  No task titles, no output screenshots, poor  formatting. |  |
| **Ability to Code Required Class/Classes** | Able to code required class, use objects effectively, and produces desired results. | Able to code required class, use objects effectively, and produces most results. | Able to code required class, somewhat use of objects, and some results are produced. | Unable to code required class or unable to use objects. |  |
| **Compilation, Execution, and Results** | Program compiles with no errors and no warnings.  Executes without errors, excellent user prompts, good use of symbols, and spacing in output.  Thorough and organized testing has  been completed and output from test cases is included. | Program compiles with no errors and some warnings. Executes without errors.  User prompts are understandable, minimum use of symbols or spacing in output. Most of the testing has been  completed. | Program compiles with no errors and lots of warnings.  Executes without errors.  User prompts are understandable, minimum use of symbols or spacing in output.  Some testing has been  completed. | Program fails to compile. Does not execute due to errors.  User prompts are misleading or non‐ existent.  No testing has been d. |  |

**Marks**: /4 =

**Teacher Remarks and Signature:**

Department of Computer Systems Eng. UET Peshawar Prepared By: Engr. Sumayyea Salahuddin.