**DEPARTMENT OF COMPUTER SCIENCE CUI,VEHARI CAMPUS**

**Assignment No: 01**

**Data Structures and Algorithms-Lab:**

****

Submitted to: **Mam Yasmeen Jana**

Submitted by: **Aqsa Hussain**

Registration No: **(SP22-BCS-083)**

Section: **(B)**

**Program No:1**

#include <iostream>

using namespace std;

int main() {

// Program 1: Basic Pointer

int x = 10;

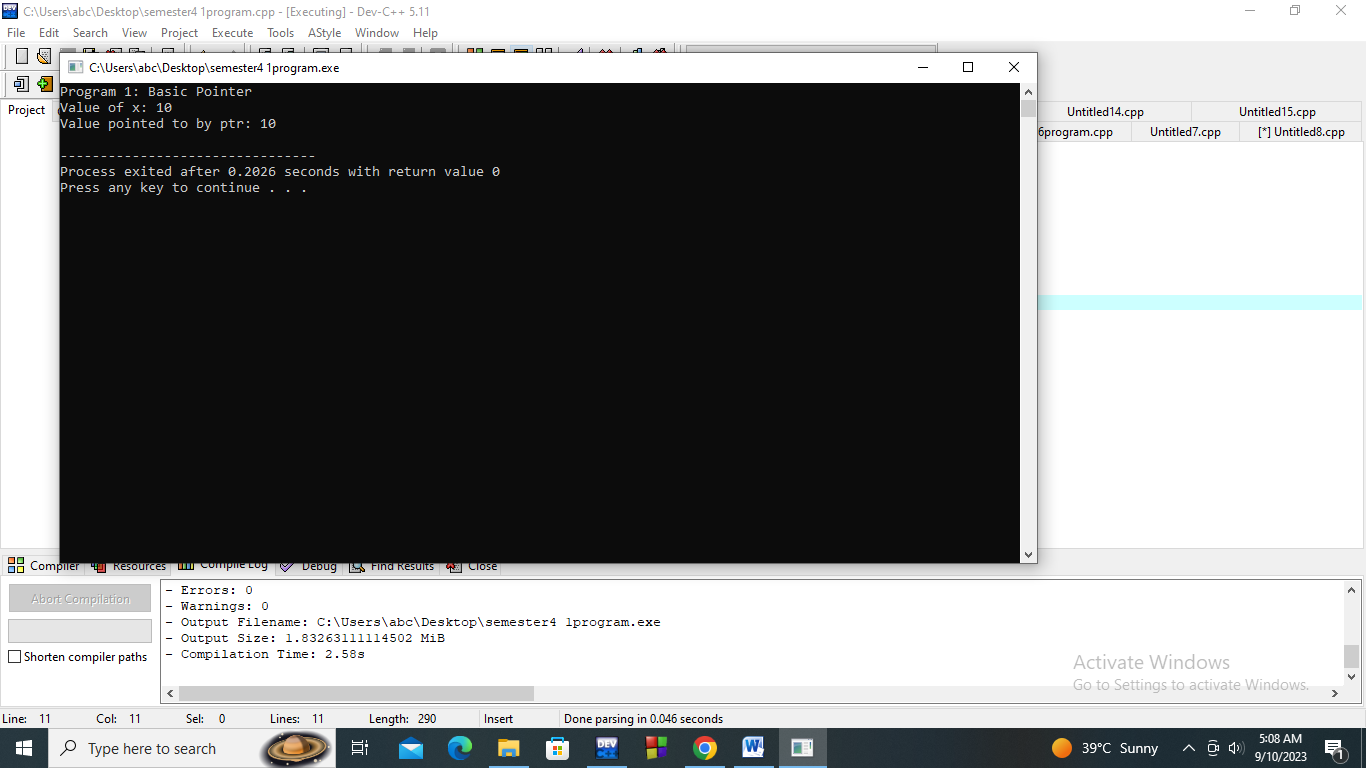
int\* ptr = &x;

cout << "Program 1: Basic Pointer" << endl;

cout << "Value of x: " << x << endl;

cout << "Value pointed to by ptr: " << \*ptr << endl;

return 0;}



**Program No:2**

#include <iostream>

using namespace std;

int main() {

int arr[] = {1, 2, 3, 4, 5};

int\* p = arr;

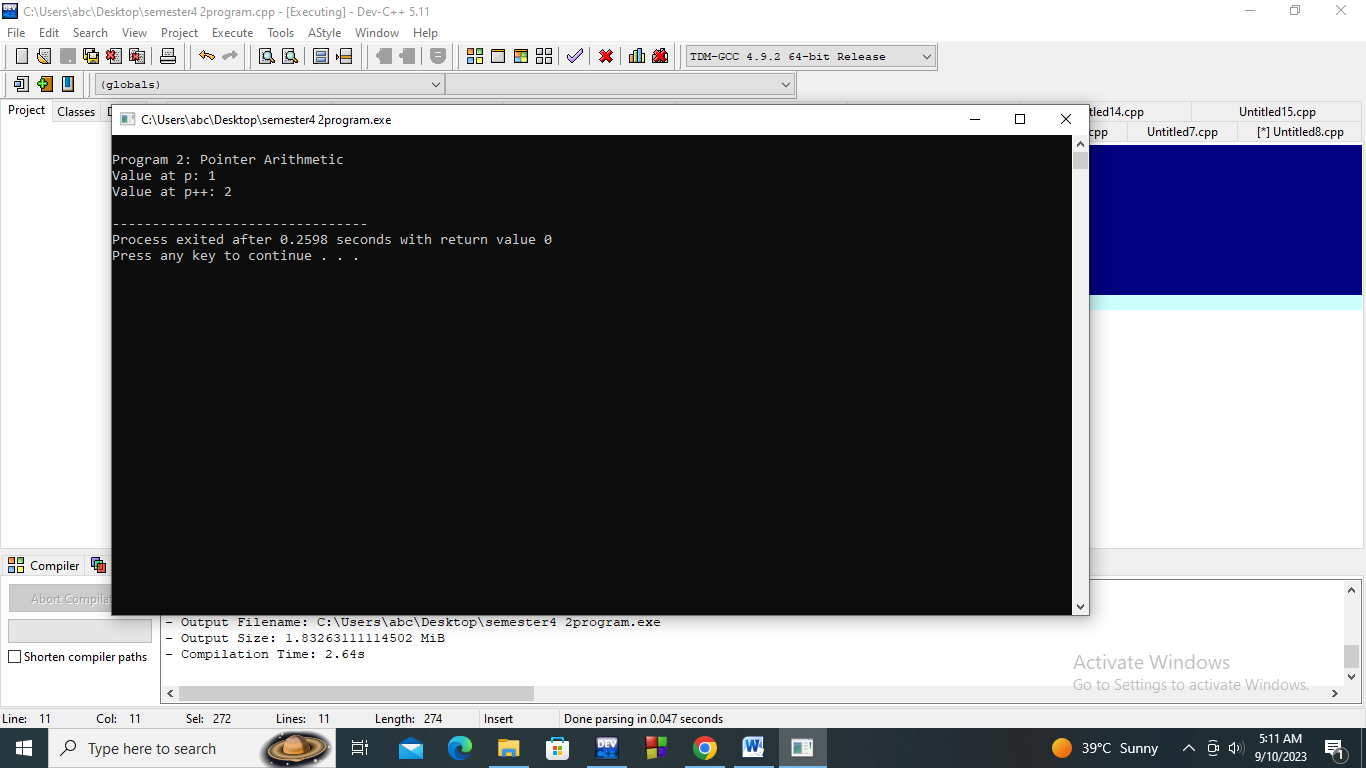
cout << "\nProgram 2: Pointer Arithmetic" << endl;

cout << "Value at p: " << \*p << endl;

p++;

cout << "Value at p++: " << \*p << endl;

return 0;}

****

**Program No: 3**

#include <iostream>

using namespace std;

int main() {

int\* dynPtr = new int;

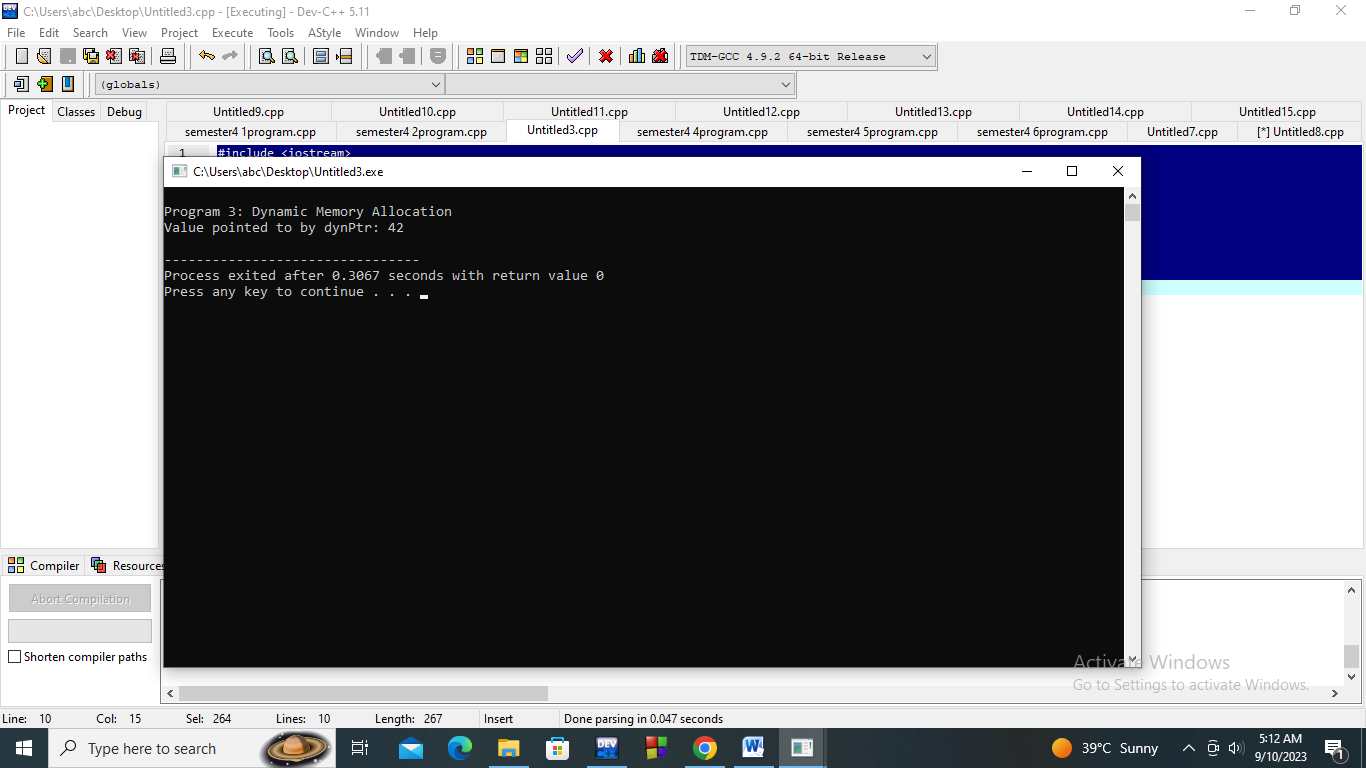
\*dynPtr = 42;

cout << "\nProgram 3: Dynamic Memory Allocation" << endl;

cout << "Value pointed to by dynPtr: " << \*dynPtr << endl;

delete dynPtr;

return 0;}



**Program No:4**

#include <iostream>

using namespace std;

int main() {

int numbers[] = {1, 2, 3, 4, 5};

int\* arrPtr = numbers;

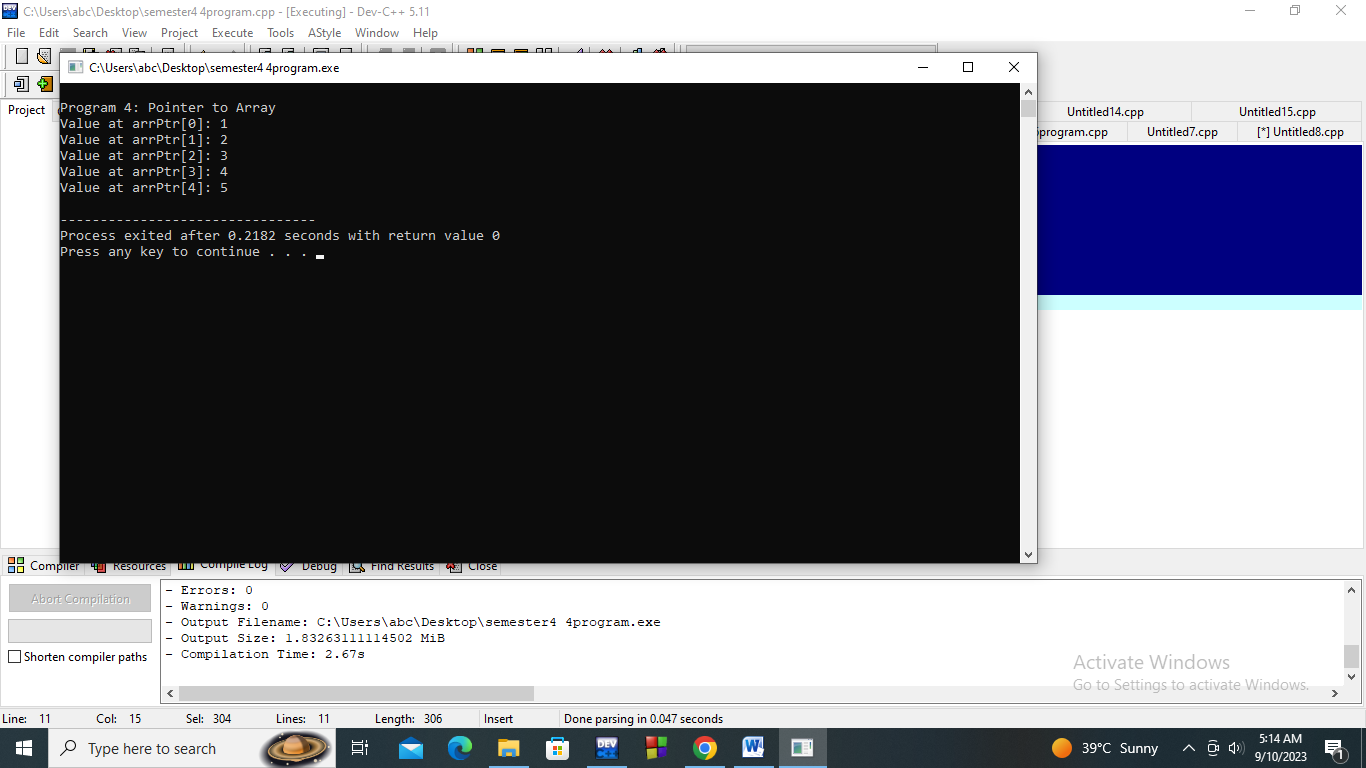
cout << "\nProgram 4: Pointer to Array" << endl;

for (int i = 0; i < 5; i++) {

cout << "Value at arrPtr[" << i << "]: " << arrPtr[i] << endl;

}

return 0;}



**Program No:5**

#include <iostream>

using namespace std;

int main() {

int y = 20;

int\* ptr1 = &y;

int\*\* ptr2 = &ptr1;

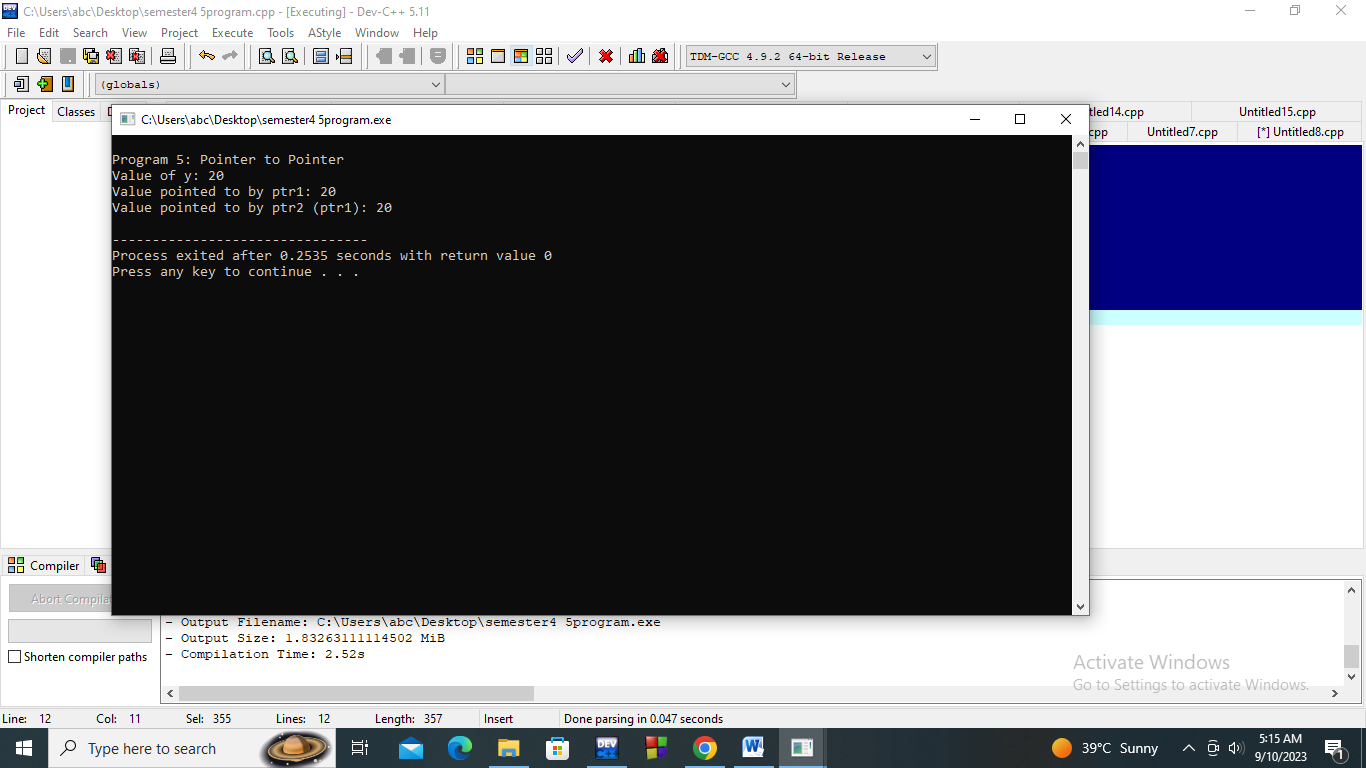
cout << "\nProgram 5: Pointer to Pointer" << endl;

cout << "Value of y: " << y << endl;

cout << "Value pointed to by ptr1: " << \*ptr1 << endl;

cout << "Value pointed to by ptr2 (ptr1): " << \*\*ptr2 << endl;

return 0;}



**Program No: 6**

#include <iostream>

using namespace std;

int main() {

const int z = 30;

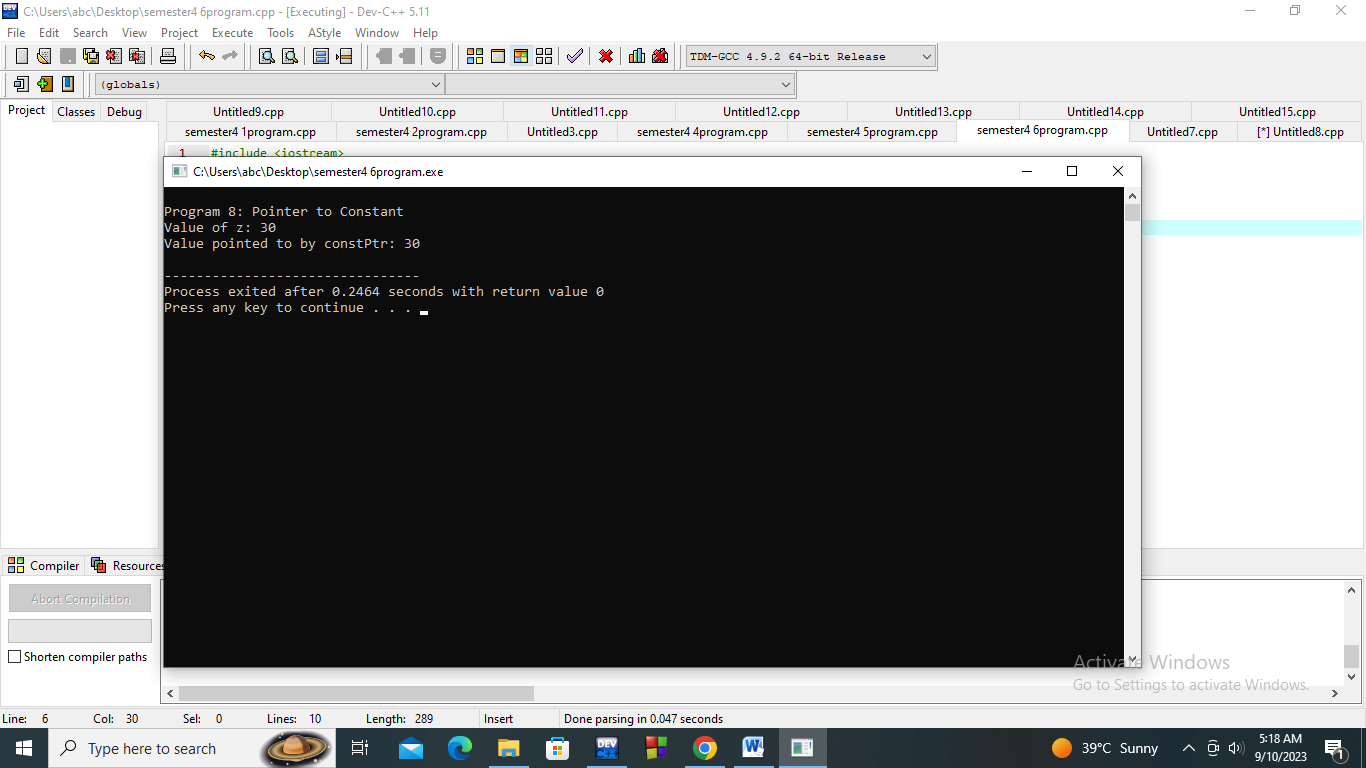
const int\* constPtr = &z;

cout << "\nProgram 8: Pointer to Constant" << endl;

cout << "Value of z: " << z << endl;

cout << "Value pointed to by constPtr: " << \*constPtr << endl;

return 0;}



**Program No:7**

#include <iostream>

using namespace std;

int main() {

int const x1 = 100;

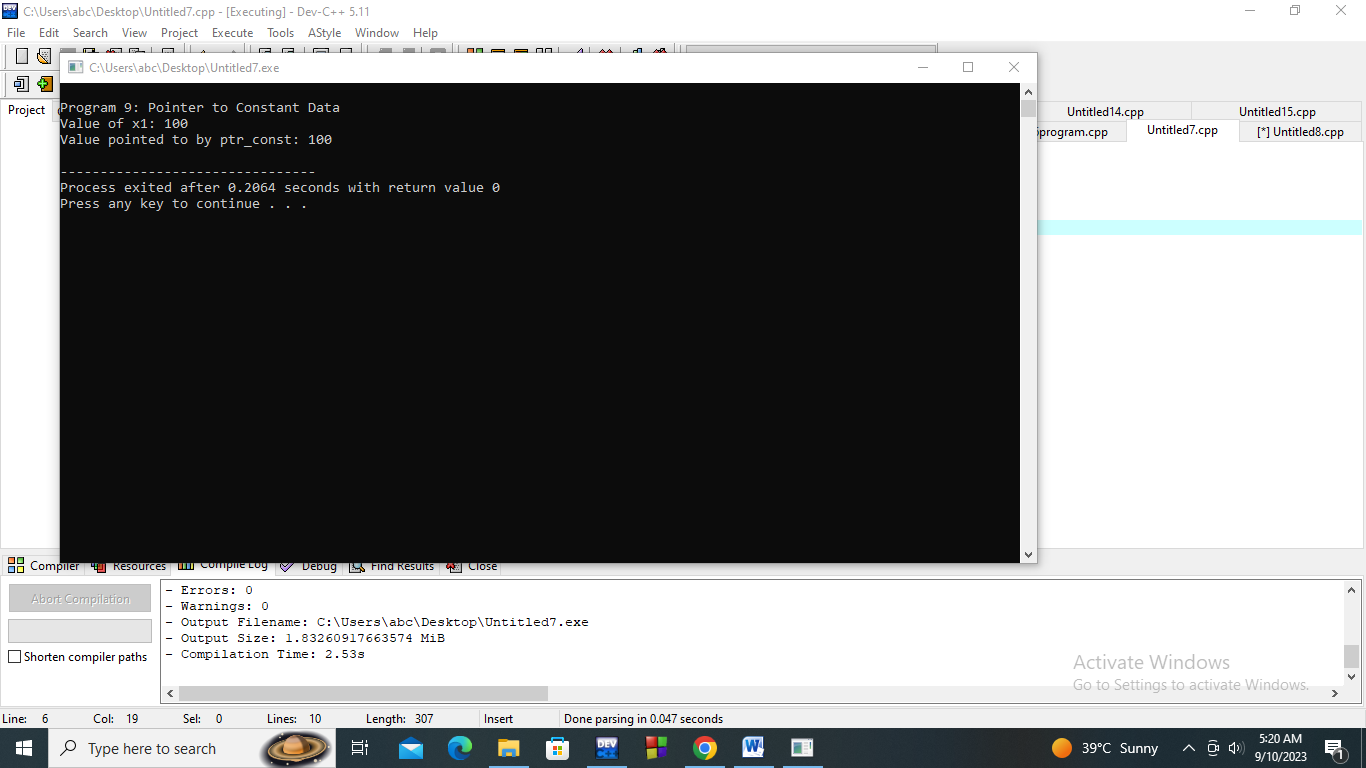
int const \*ptr\_const = &x1;

cout << "\nProgram 9: Pointer to Constant Data" << endl;

cout << "Value of x1: " << x1 << endl;

cout << "Value pointed to by ptr\_const: " << \*ptr\_const << endl;

return 0;}



**Program No:8**

include <iostream>

using namespace std;

int mai#n() {

int v = 15;

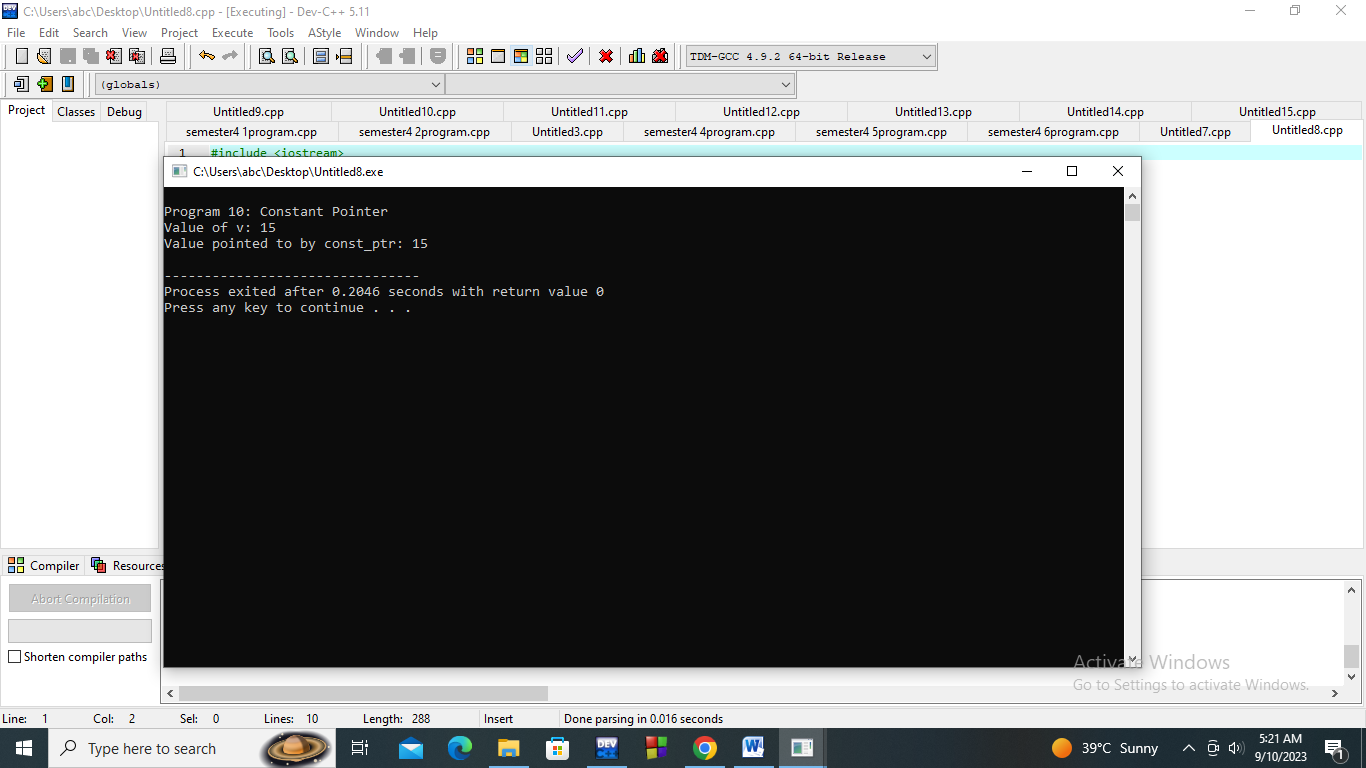
int\* const const\_ptr = &v;

cout << "\nProgram 10: Constant Pointer" << endl;

cout << "Value of v: " << v << endl;

cout << "Value pointed to by const\_ptr: " << \*const\_ptr << endl;

return 0;}



**Program No: 9**

#include <iostream>

using namespace std;

int main() {

class MyClass {

public:

int data;

MyClass(int val) : data(val) {}

};

MyClass obj(42);

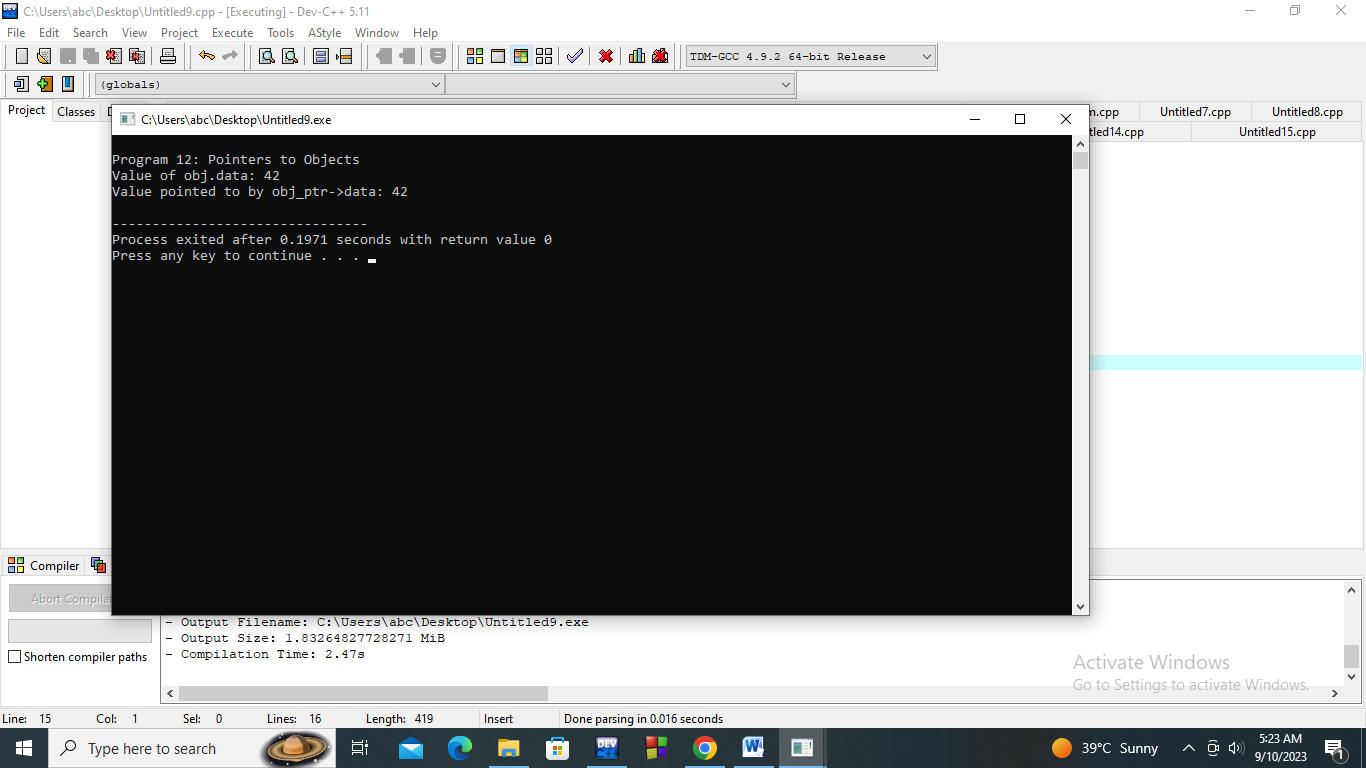
MyClass\* obj\_ptr = &obj;

cout << "\nProgram 12: Pointers to Objects" << endl;

cout << "Value of obj.data: " << obj.data << endl;

cout << "Value pointed to by obj\_ptr->data: " << obj\_ptr->data << endl;

return 0;}



**Program No:10**

#include <iostream>

using namespace std;

int main() {

class MyStruct {

public:

int value;

};

MyStruct myStruct;

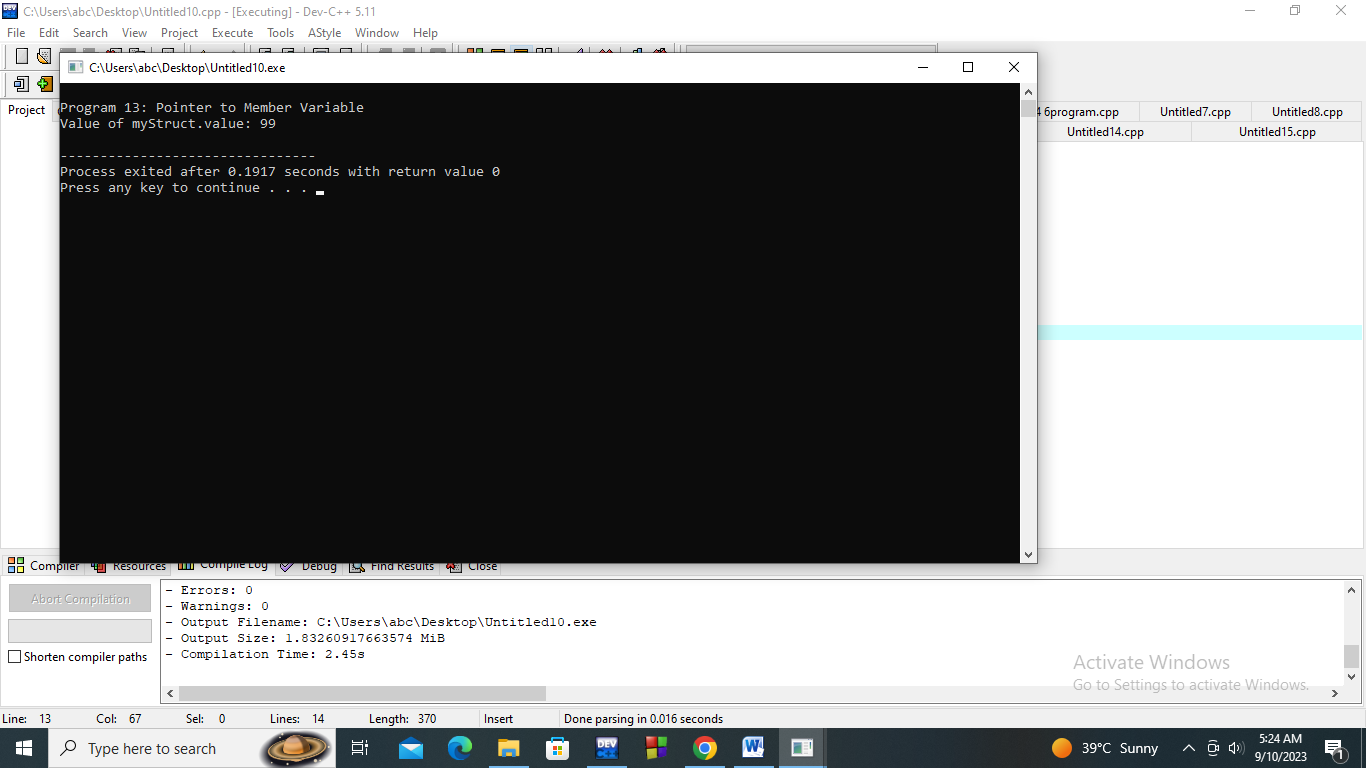
int MyStruct::\*memberPtr = &MyStruct::value;

cout << "\nProgram 13: Pointer to Member Variable" << endl;

myStruct.\*memberPtr = 99;

cout << "Value of myStruct.value: " << myStruct.value << endl;

return 0;}



**Program No:11**

#include <iostream>

using namespace std;

int main() {

class MyClass2 {

public:

int add(int a, int b) { return a + b; }

};

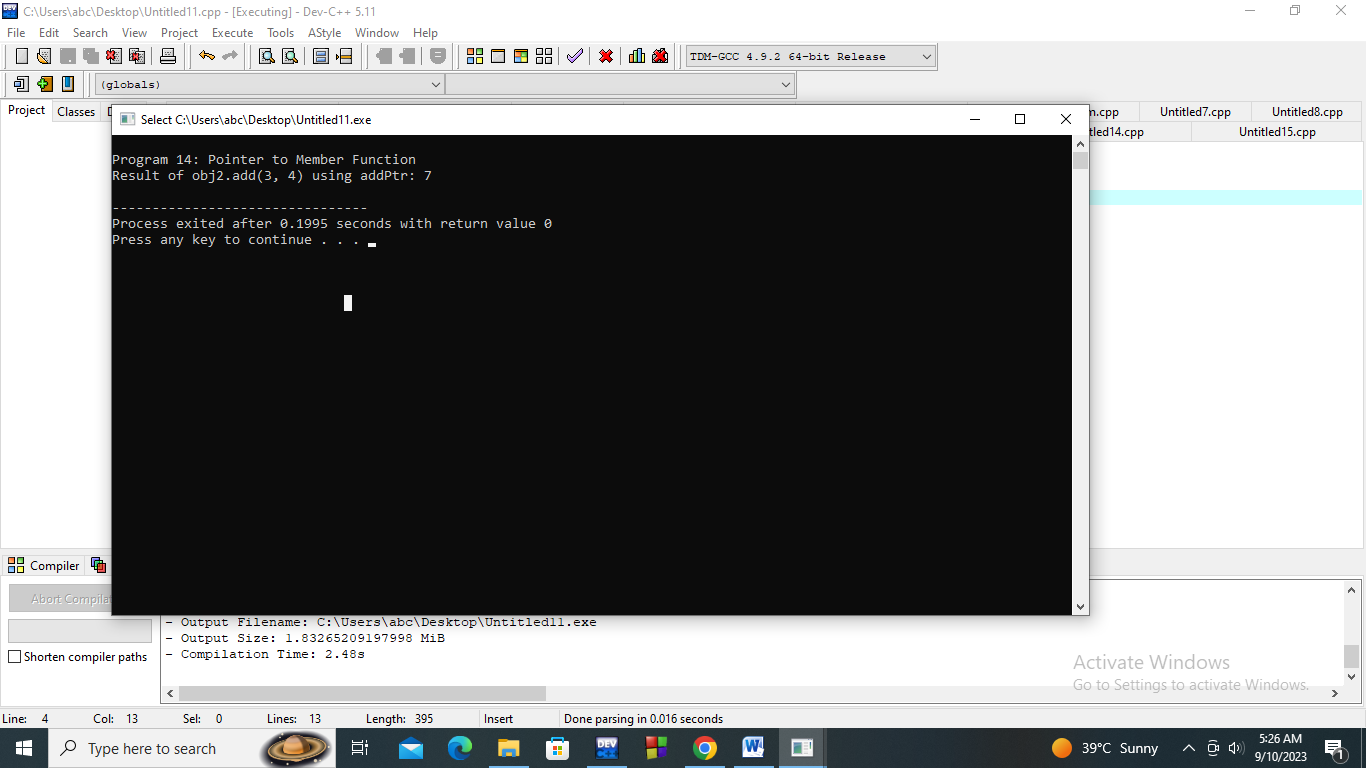
int (MyClass2::\*addPtr)(int, int) = &MyClass2::add;

MyClass2 obj2;

cout << "\nProgram 14: Pointer to Member Function" << endl;

cout << "Result of obj2.add(3, 4) using addPtr: " << (obj2.\*addPtr)(3, 4) << endl;

return 0;}



**Program No:12**

#include <iostream>

using namespace std;

int main() {

int arr15[] = {100, 200, 300};

int\* ptrArr[3];

for (int i = 0; i < 3; i++) {

ptrArr[i] = &arr15[i];

}

cout << "\nProgram 15: Pointer to Array of Pointers" << endl;

for (int i = 0; i < 3; i++) {

cout << "Value pointed to by ptrArr[" << i << "]: " << \*ptrArr[i] << endl;

}

return 0;

}

void printValue(int\* ptr) {

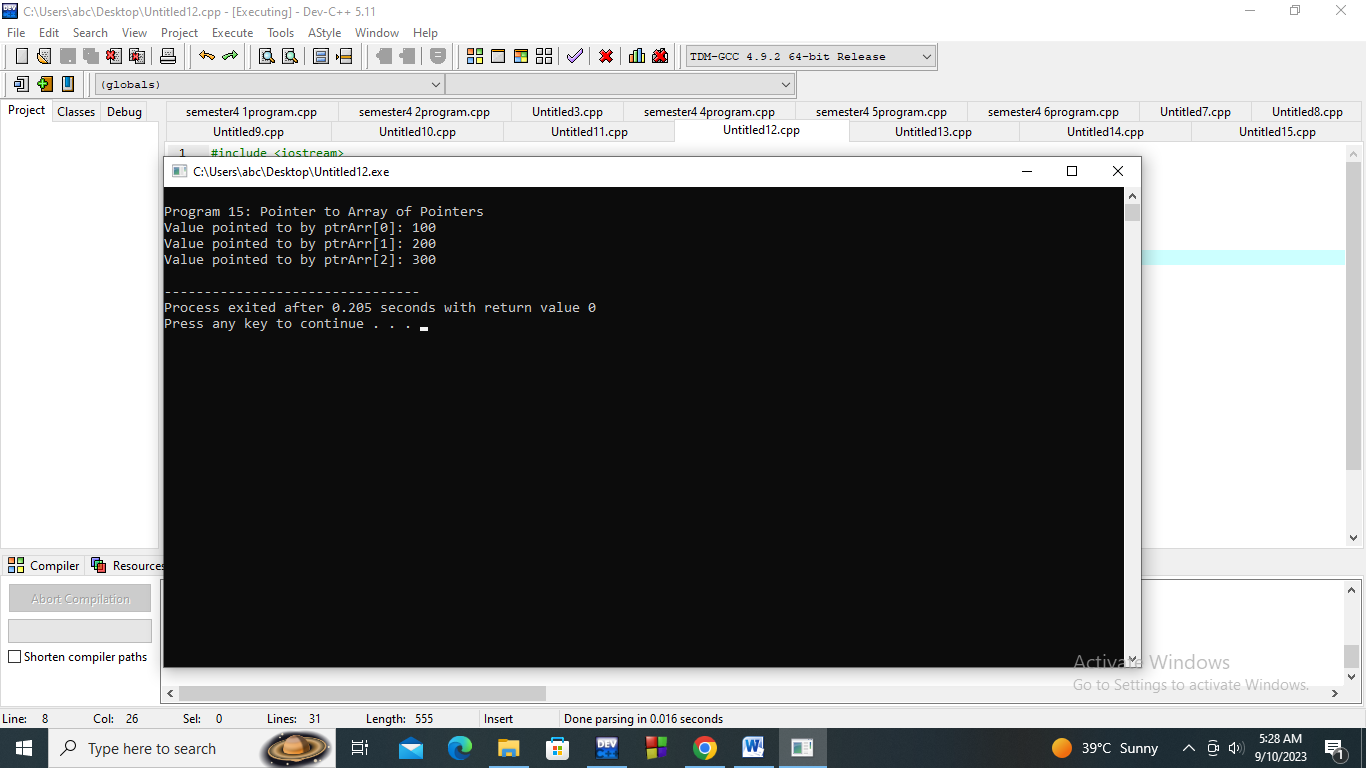
cout << "Value pointed to by ptr: " << \*ptr << endl;

}

int add(int a, int b) {

return a + b;

}



**Program No:13**

#include <iostream>

void modifyValue(int \*ptr) {

(\*ptr)++;

}

int main() {

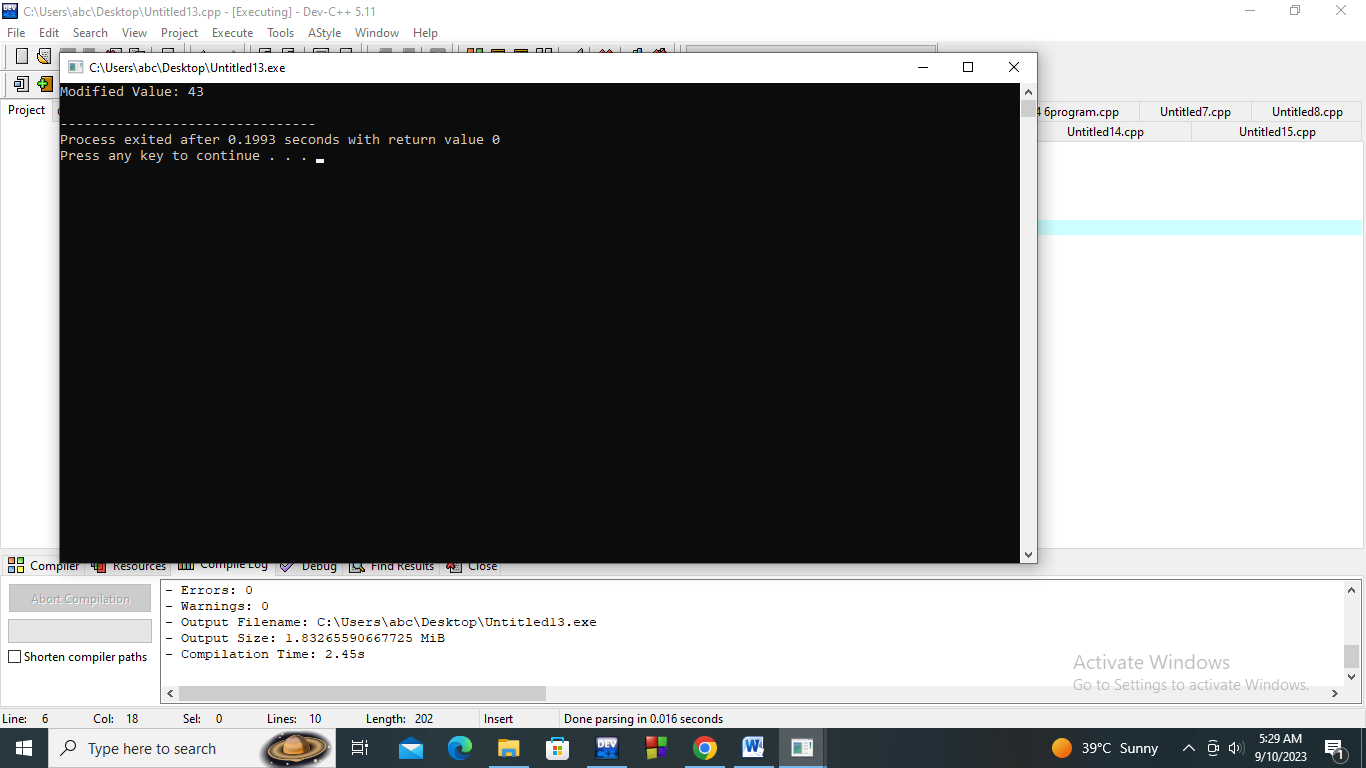
int num = 42;

modifyValue(&num);

std::cout << "Modified Value: " << num << std::endl;

return 0;

}



**Program No:14**

#include <iostream>

class MyClass {

public:

int data = 42;

void printData() {

std::cout << "Data: " << data << std::endl;

}

};

int main() {

MyClass obj;

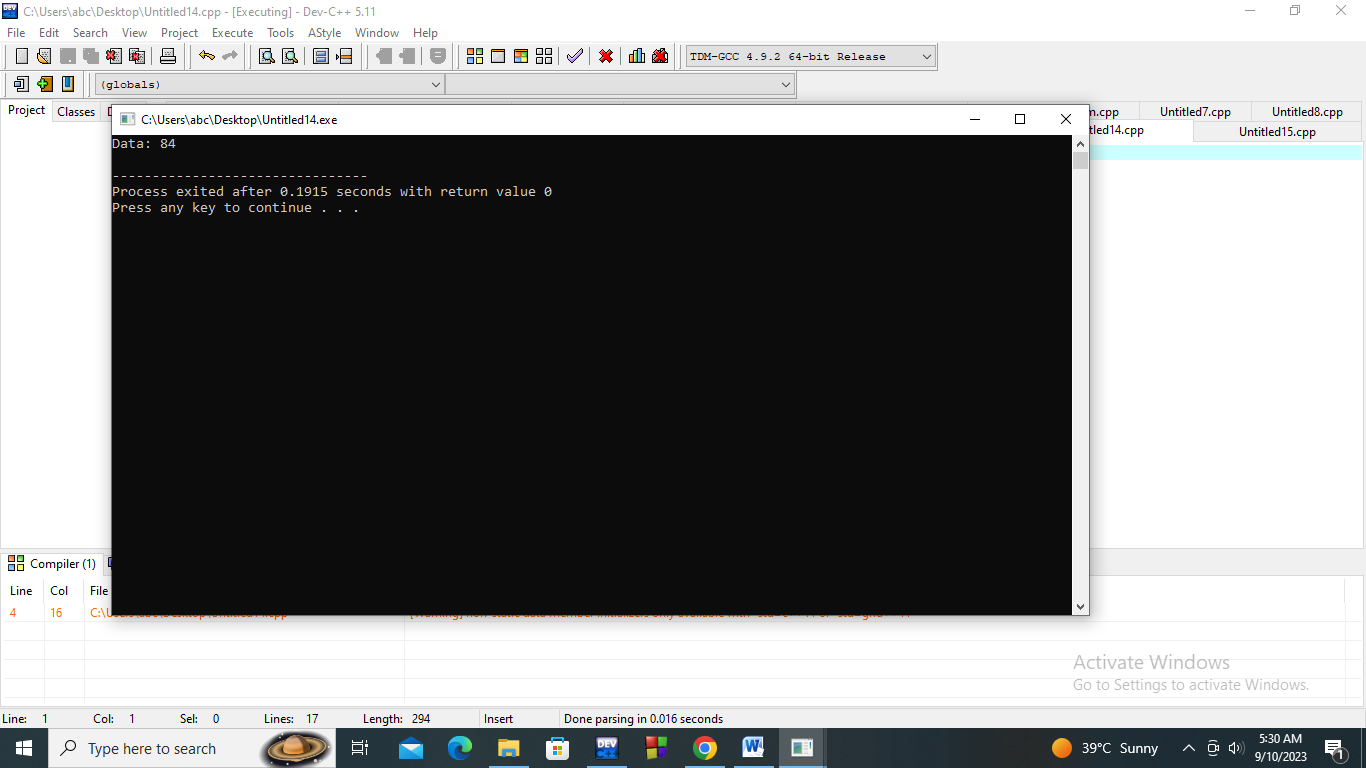
int MyClass::\*ptr = &MyClass::data;

(obj.\*ptr) = 84;

obj.printData();

return 0;

}



**Program No:15**

#include <iostream>

class Math {

public:

int add(int a, int b) {

return a + b;

}

};

int main() {

Math math;

int (Math::\*ptr)(int, int) = &Math::add;

std::cout << "Sum: " << (math.\*ptr)(3, 4) << std::endl;

return 0;

}

