Comsats University Islamabad, Vehari Campus

Department: Computer Science

Lab activity 1

Program: BSCS

Submitted by:

Ayesha Mushtaq **SP22-BCS-102**

Subject: DSA-LAB

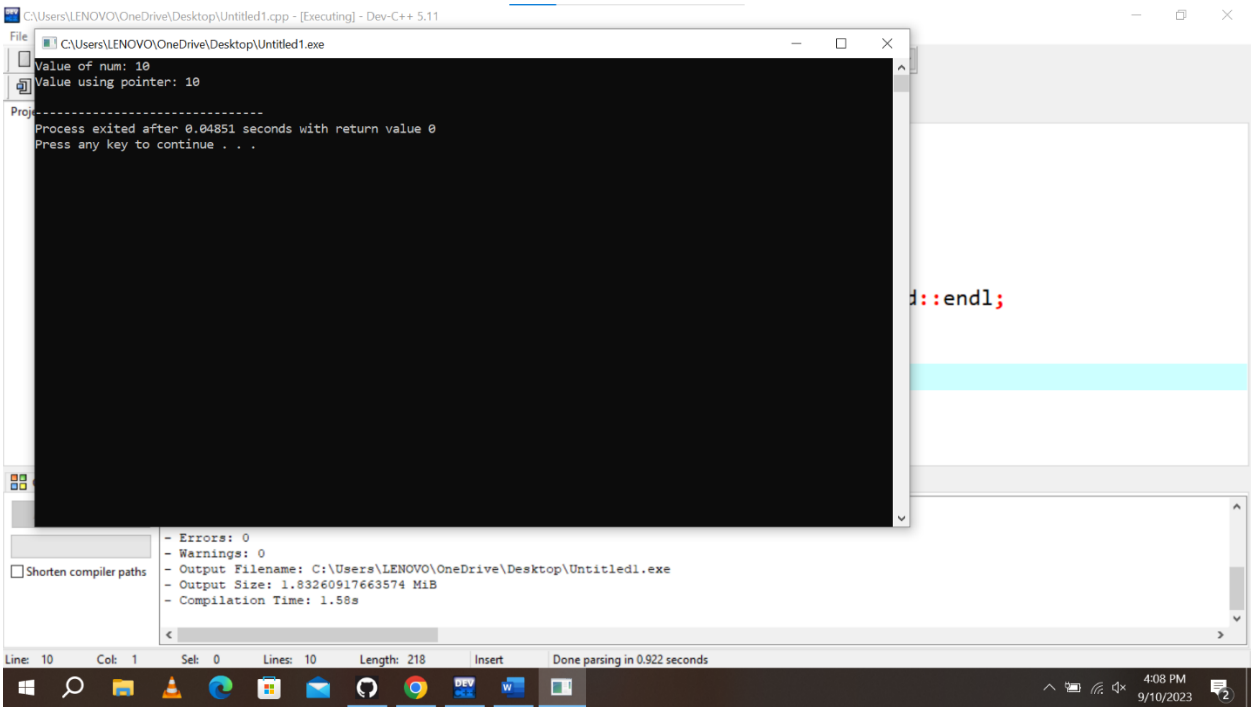
Semester: 4th

Section: B

Submitted to: mam Yasmeen jana

Q#1: Basic Pointer Declaration and Usage

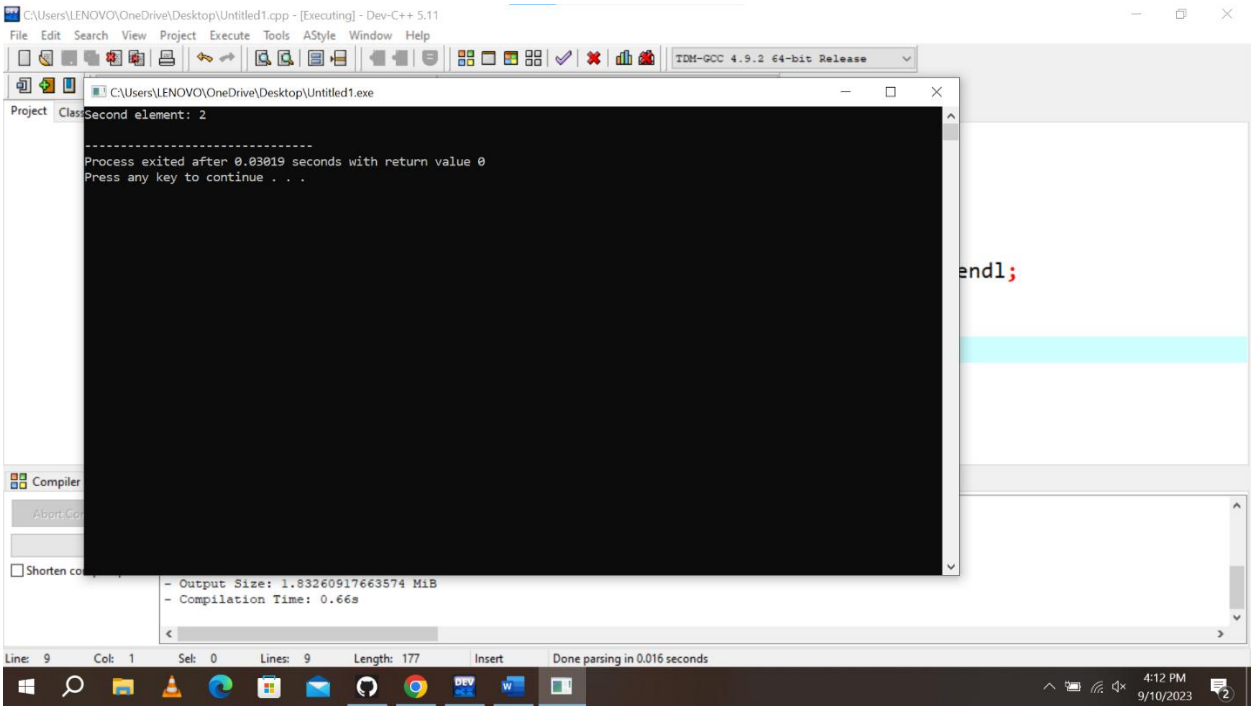
```
#include <iostream>


int main() {
    int num = 10;
    int *ptr = &num;
    std::cout << "Value of num: " << num << std::endl;
    std::cout << "Value using pointer: " << *ptr << std::endl;
    return 0;
}
```



Q#2: Pointer Arithmetic

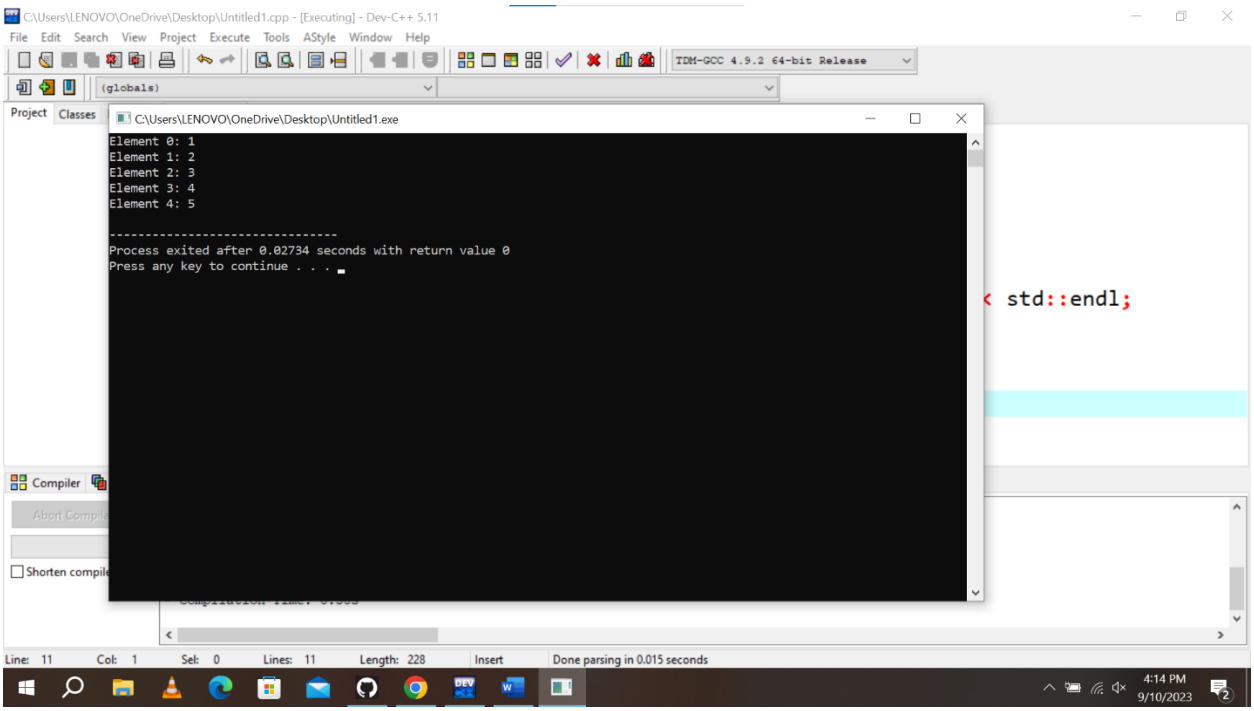
```
#include <iostream>


int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;
    std::cout << "Second element: " << *(ptr + 1) << std::endl;
```

```
    return 0;
}
```



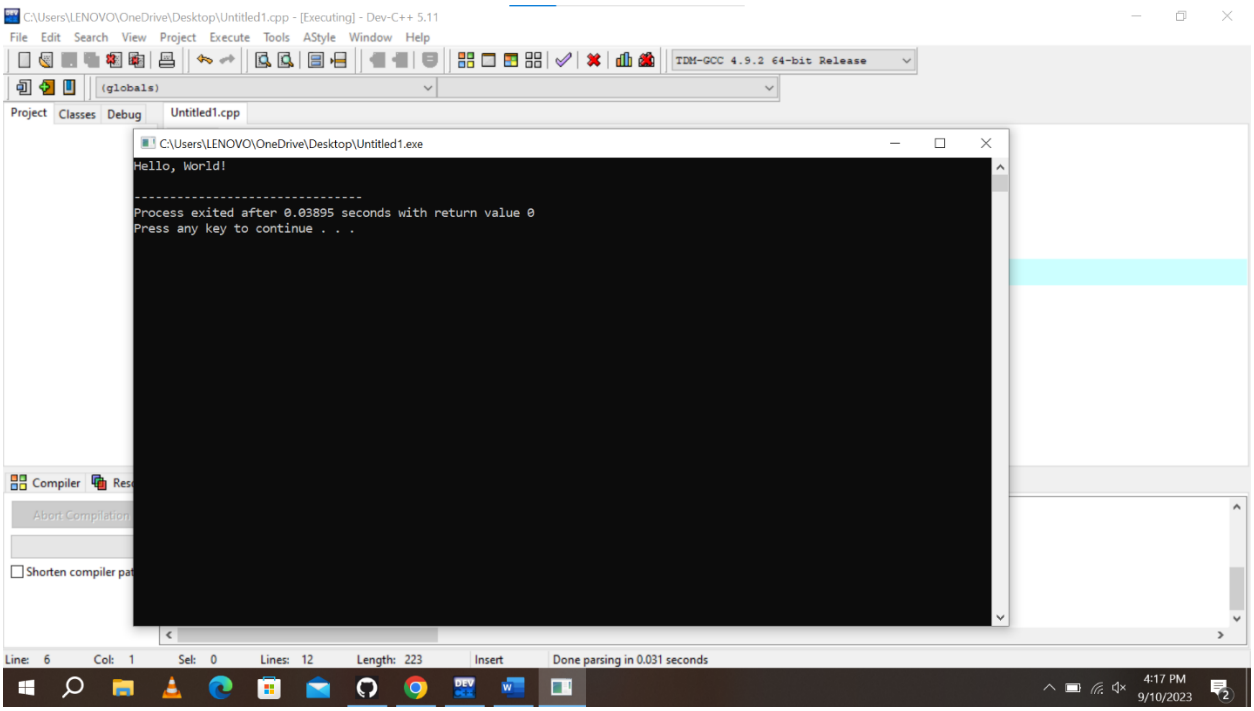
Q#3: pointer to array

```
#include <iostream>


int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;
    for (int i = 0; i < 5; ++i) {
        std::cout << "Element " << i << ": " << *(ptr + i) << std::endl;
    }
    return 0;
}
```



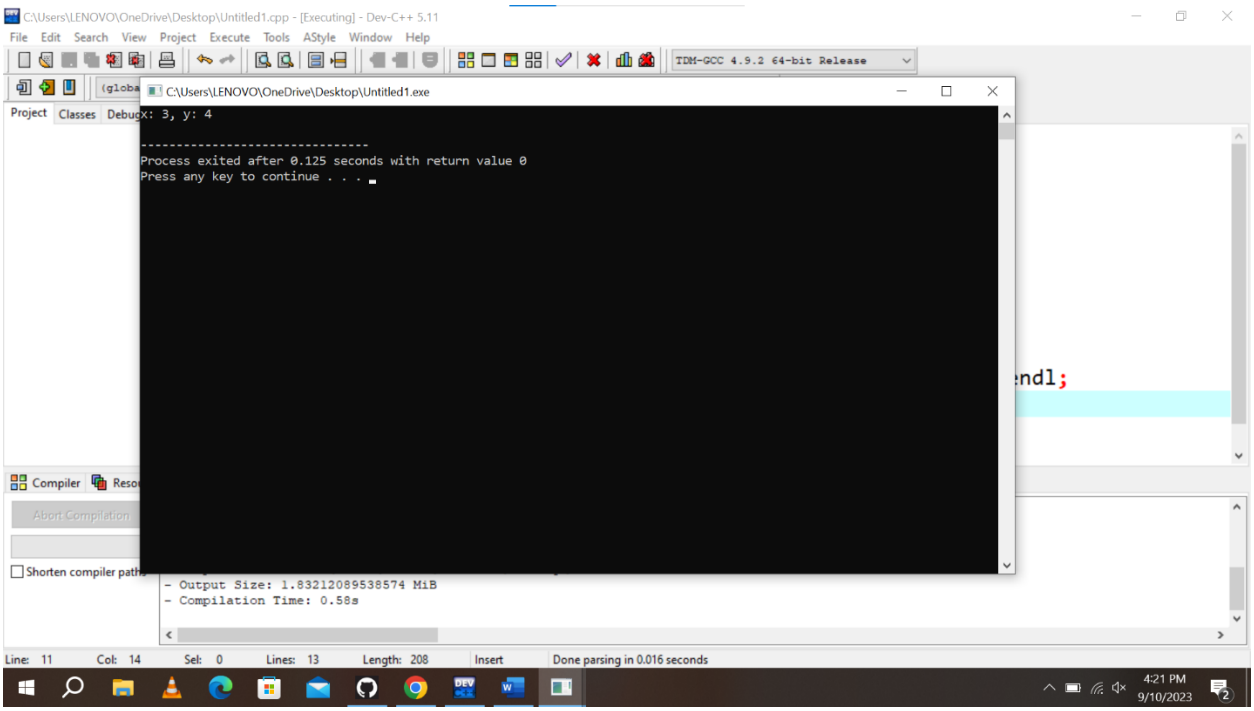
Q#4: pointers to function

```
#include <iostream>


void sayHello() {
    std::cout << "Hello, World!" << std::endl;
}


int main() {
    void (*func_ptr)() = &sayHello;
    (*func_ptr)(); // Call the function using a pointer
    return 0;
}
```



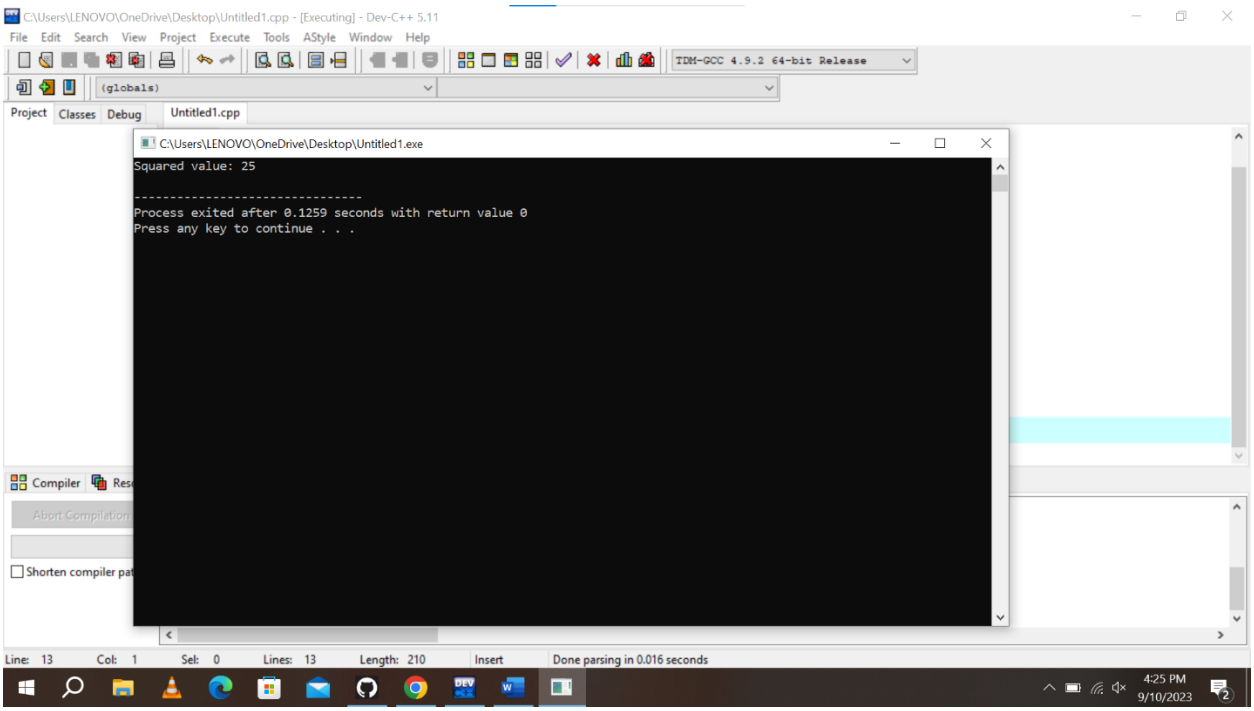
Q#5: pointers to structure

```
#include <iostream>


struct Point {
    int x, y;
};


int main() {
    Point p = {3, 4};
    Point *ptr = &p;
    std::cout << "x: " << ptr->x << ", y: " << ptr->y << std::endl;
    return 0;
}
```



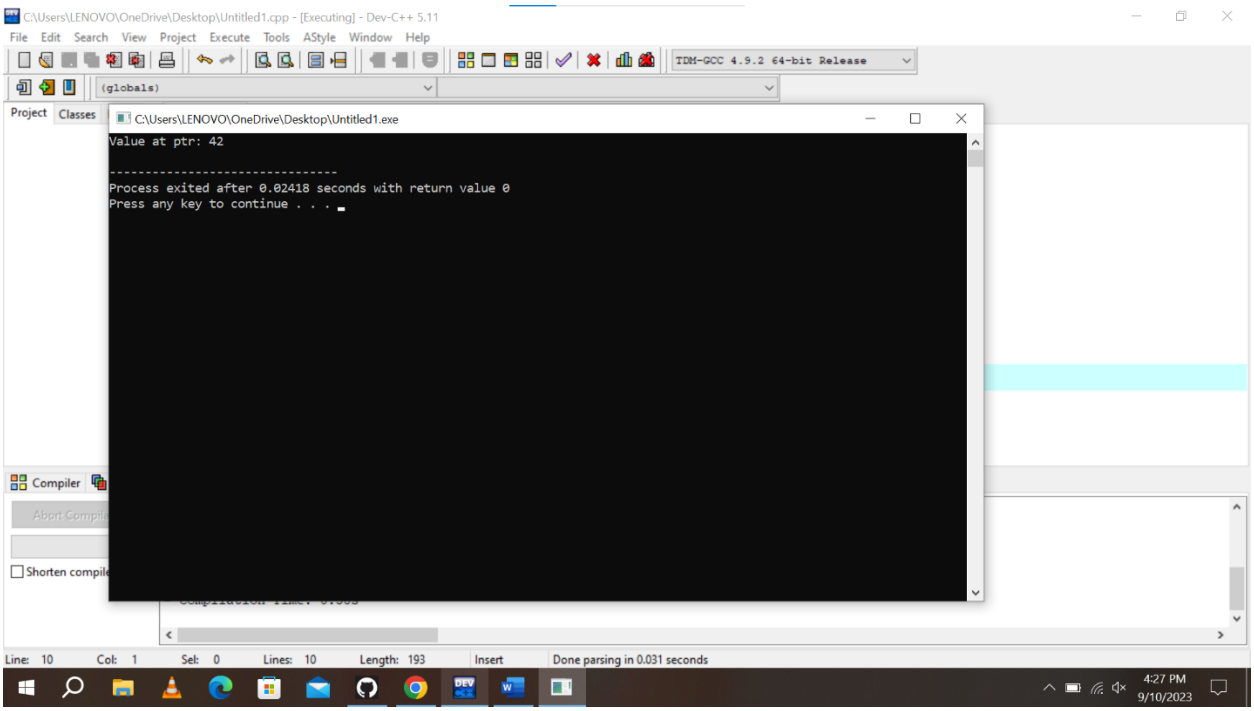
Q#6: pointers as functional parameters

```
#include <iostream>


void square(int *ptr) {
    *ptr = (*ptr) * (*ptr);
}


int main() {
    int num = 5;
    square(&num);
    std::cout << "Squared value: " << num << std::endl;
    return 0;
}
```



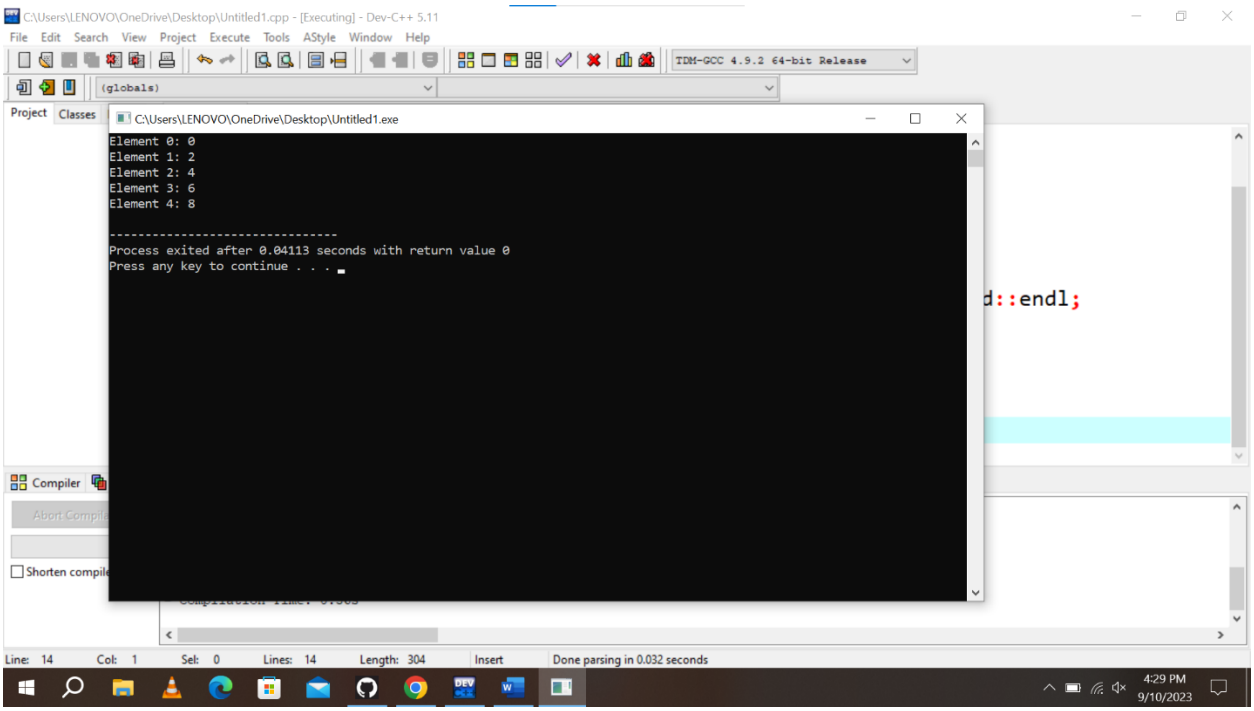
Q#7: Dynamic Memory Allocation with new and delete

```
#include <iostream>


int main() {
    int *ptr = new int;
    *ptr = 42;
    std::cout << "Value at ptr: " << *ptr << std::endl;
    delete ptr; // Deallocate memory
    return 0;
}
```



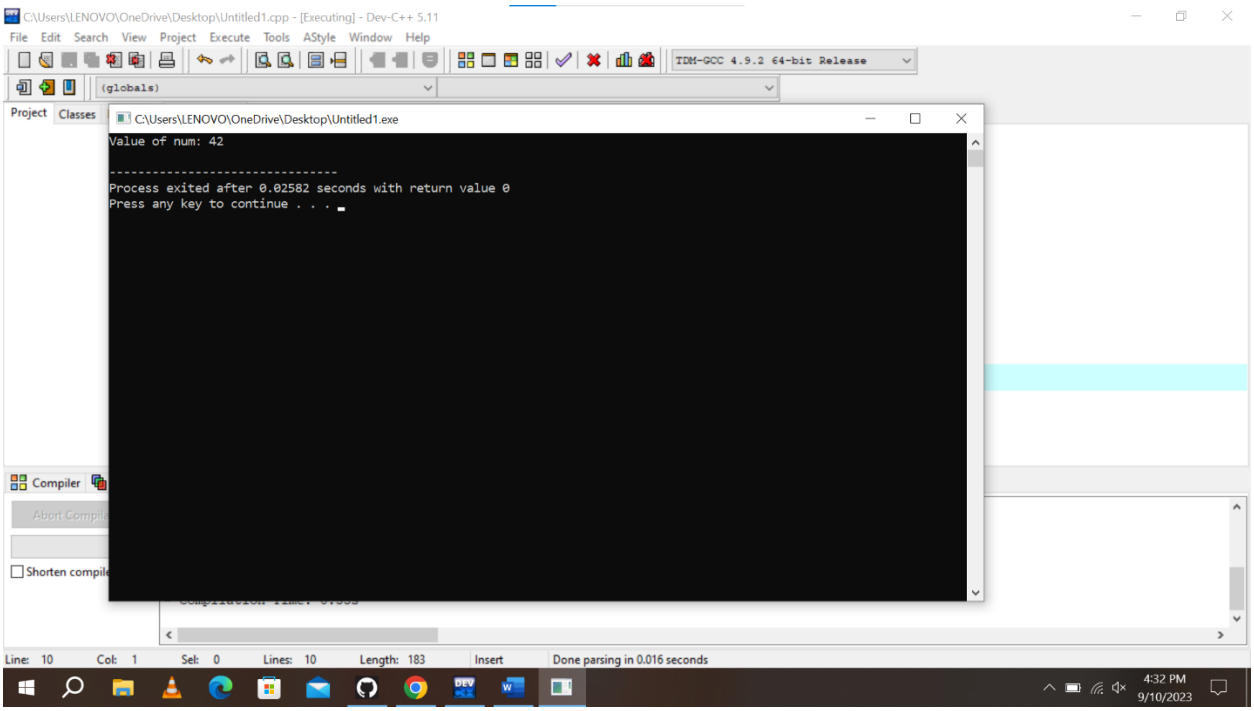
Q#8: arrays and pointers:

```
#include <iostream>


int main() {
   int *arr = new int[5];
   for (int i = 0; i < 5; ++i) {
      arr[i] = i * 2;
   }
   for (int i = 0; i < 5; ++i) {
      std::cout << "Element " << i << ": " << arr[i] << std::endl;
   }
   delete[] arr; // Deallocate memory
   return 0;
}
```



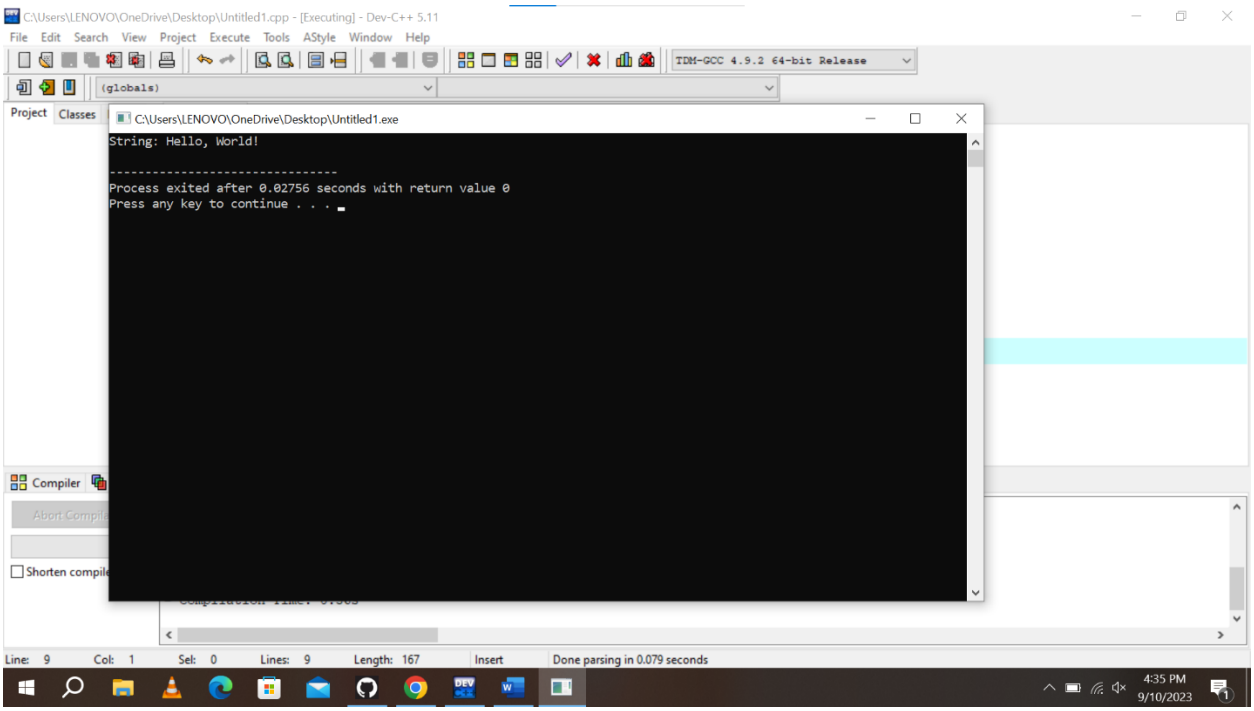
Q#9: Pointer to Pointer (Double Pointer)

```
#include <iostream>


int main() {
    int num = 42;
    int *ptr1 = &num;
    int **ptr2 = &ptr1;
    std::cout << "Value of num: " << **ptr2 << std::endl;
    return 0;
}
```



Q#10: pointers and string

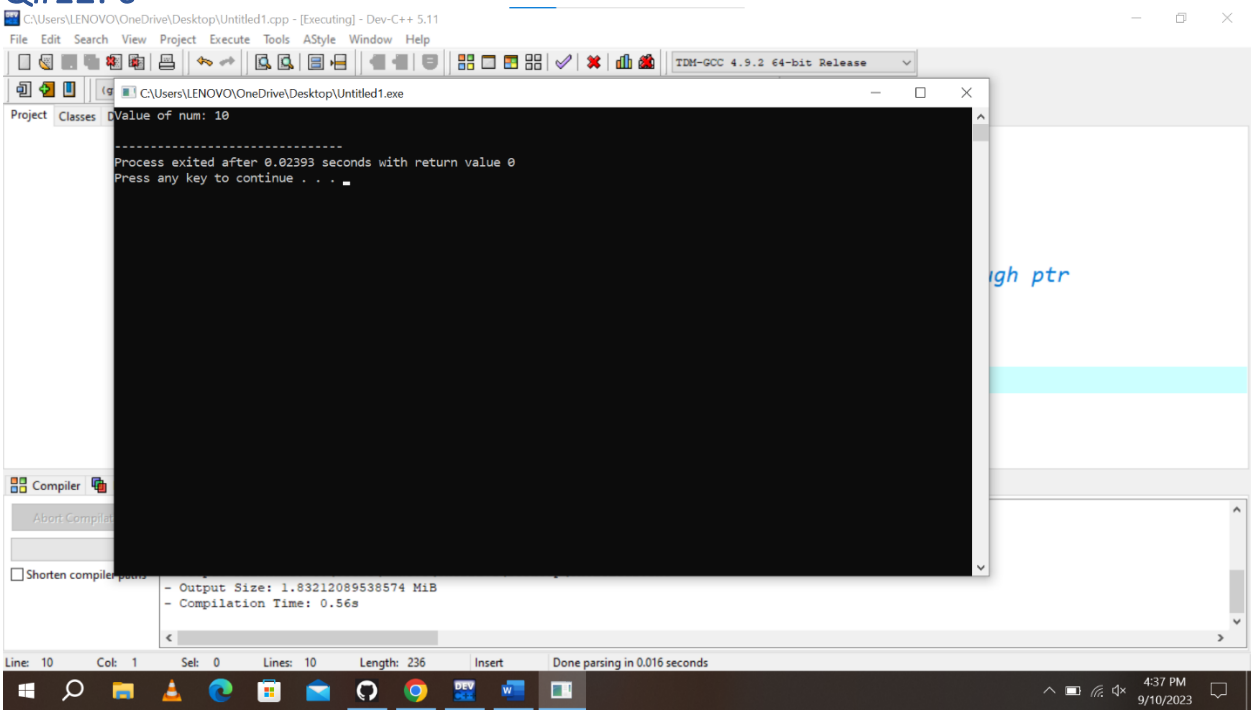
```
#include <iostream>

#include <cstring>


int main() {

   const char *str = "Hello, World!";

   std::cout << "String: " << str << std::endl;

   return 0;

}
```



Q#11: pointers to constant data

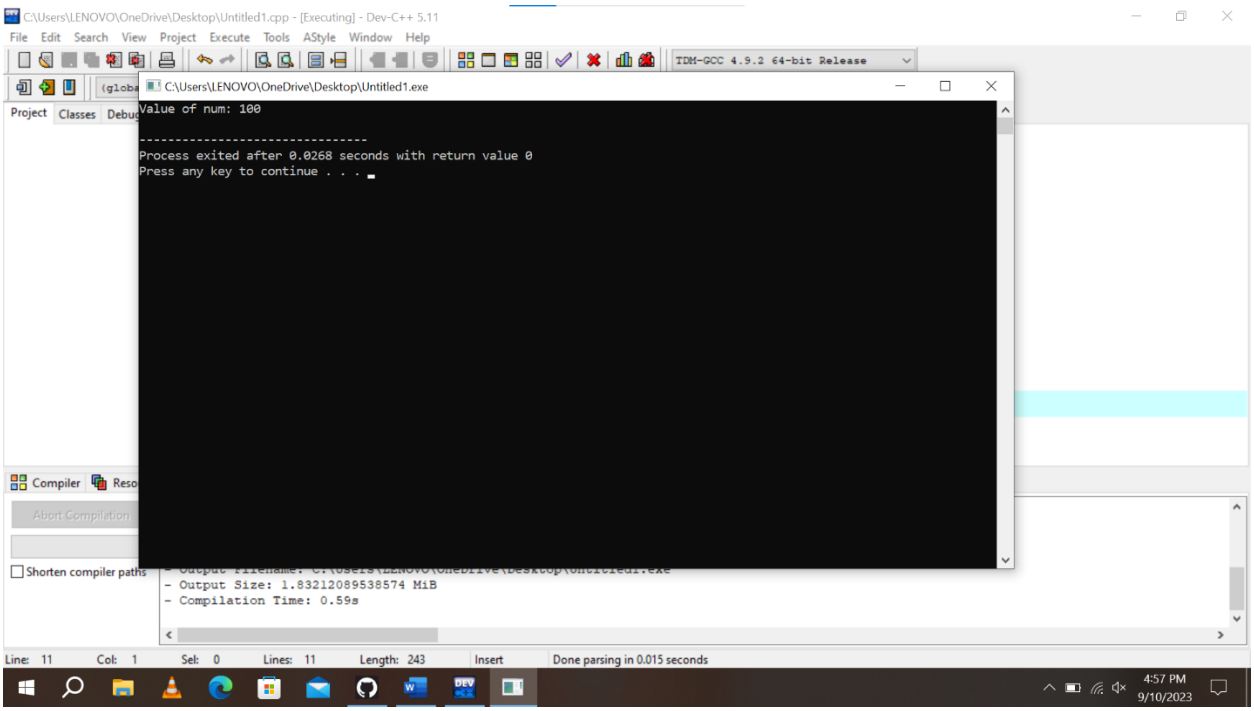
```
#include <iostream>


int main() {
    const int num = 10;
    const int *ptr = &num;
    // *ptr = 20; // Error: Cannot modify constant data through ptr
    std::cout << "Value of num: " << *ptr << std::endl;
    return 0;
}
```

Q#12: c



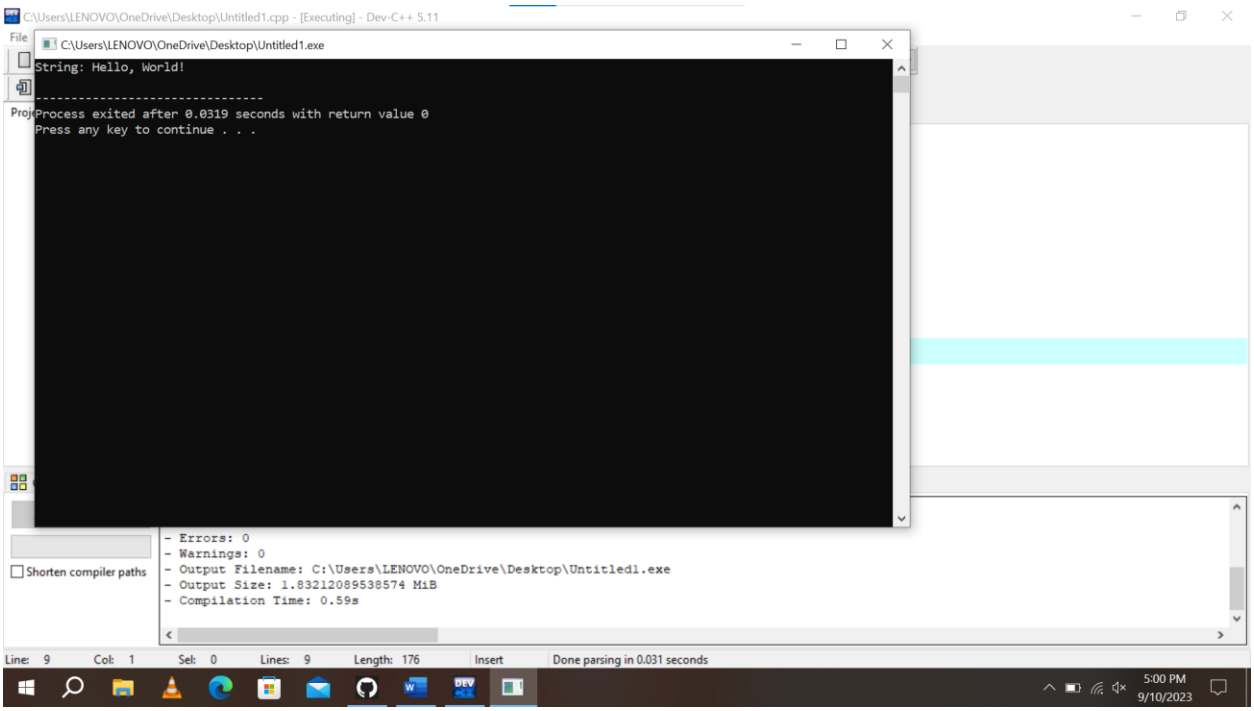
onstant pointer

```
#include <iostream>


int main() {
    int num = 42;
    int *const ptr = &num;
    *ptr = 100; // OK
    // ptr = nullptr; // Error: Cannot change the pointer
    std::cout << "Value of num: " << *ptr << std::endl;
    return 0;
}
```



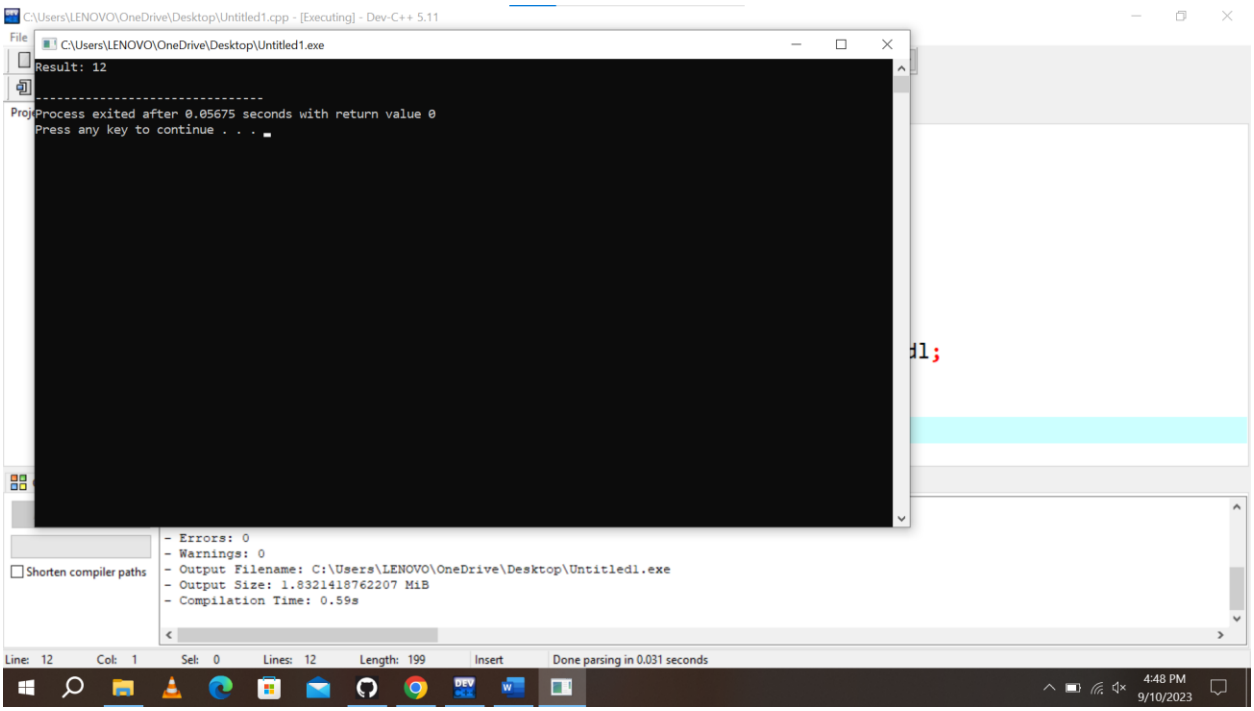
Q#13: pointer to constant string

```
#include <iostream>


int main() {
    const char str[] = "Hello, World!";
    const char *ptr = str;
    std::cout << "String: " << ptr << std::endl;
    return 0;
}
```



Q#14: pointer to constant function

```
#include <iostream>


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


int main() {
    int (*ptr)(int, int) = &add;
    std::cout << "Result: " << (*ptr)(5, 7) << std::endl;
    return 0;
}
```



Q#15: pointer to member function

```
#include <iostream>


class MyClass {

public:

   int value;


   void setValue(int val) {

      value = val;

   }

};


int main() {

   MyClass obj;

   MyClass *ptr = &obj;

   ptr->setValue(42);
```

```
    std::cout << "Value: " << obj.value << std::endl;

    return 0;
}
```