

DEPARTMENT OF COMPUTER
SCIENCE CUI, VEHARI CAMPUS

Assignments no. 01



Subject:

Data Structure & Algorithm-Lab

Submitted to:

Respected Mam Yasmeen Jana

Submitted by:

Sadaf Fatima (SP22-BCS-B24-100-B)

1. Basic Pointer Declaration and Use:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int x = 10;
```

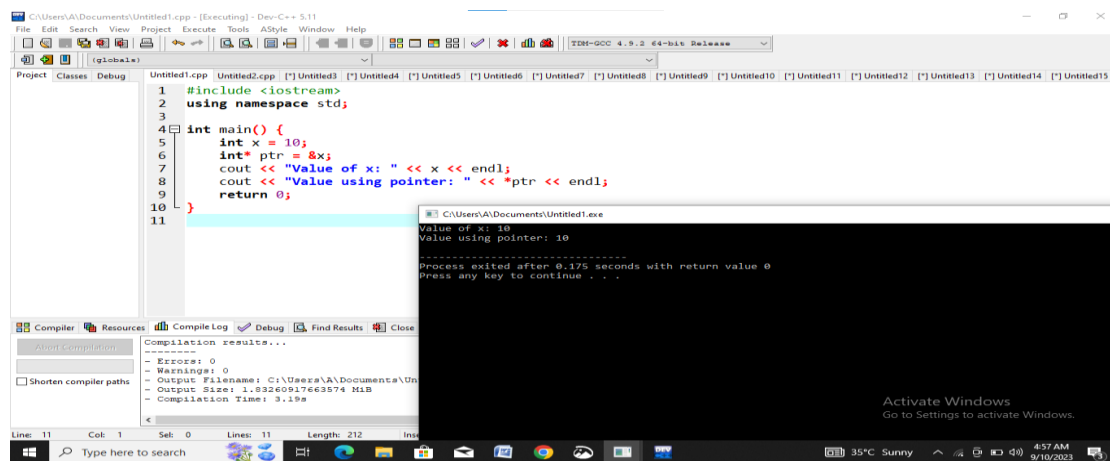
```
    int* ptr = &x;
```

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

```
    cout << "Value using pointer: " << *ptr << endl;
```

```
    return 0;
```

```
}
```



2. Pointer Arithmetic:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

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

```
    int* ptr = arr;
```

```
    cout << "Array elements using pointers: ";
```

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

```
        cout << *ptr << " ";
```

```
        ptr++;
```

```
    }
```

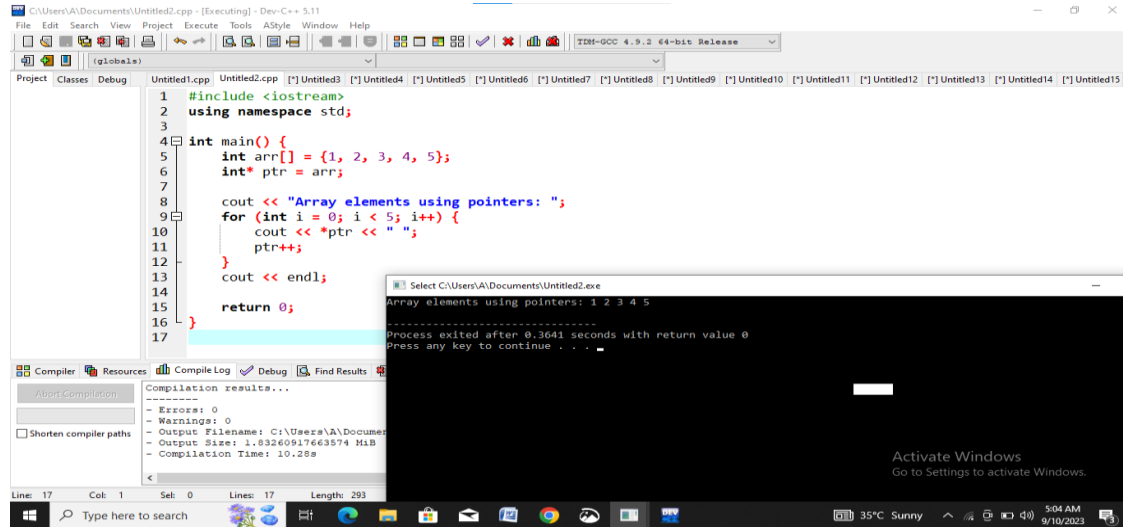
```

cout << endl;

return 0;

}

```



3. Pointer to Function:

```

#include <iostream>

using namespace std;

void printHello() {

    cout << "Hello, World!" << endl;

}

int main() {

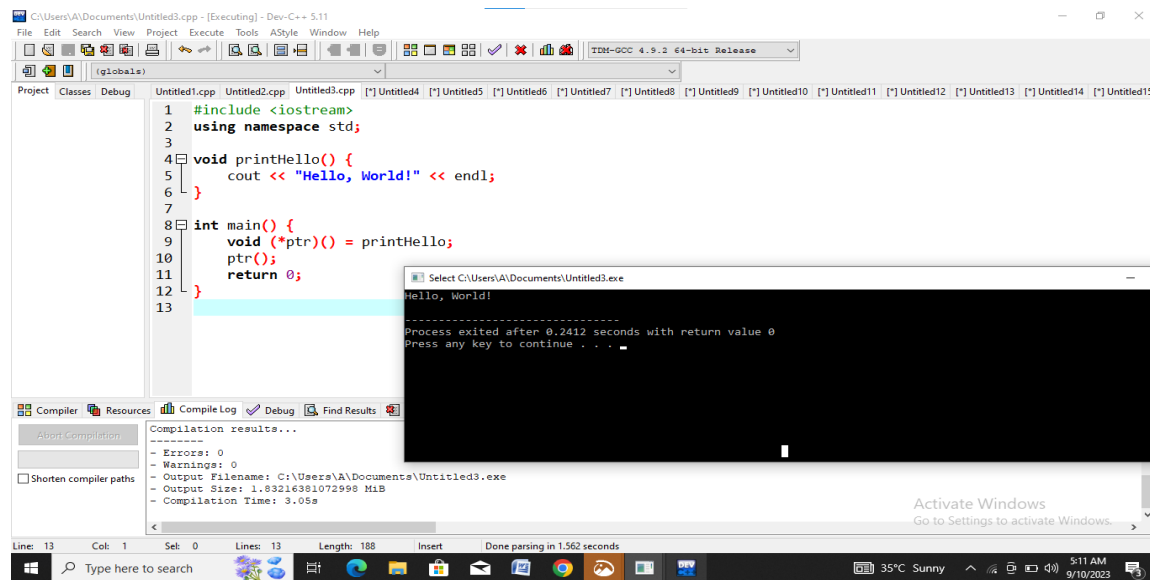
    void (*ptr)() = printHello;

    ptr();

    return 0;

}

```



4. Passing Pointers to Functions:

```
#include <iostream>
```

```
using namespace std;
```

```
void doubleValue(int* ptr) {
```

```
    *ptr *= 2;
```

```
}
```

```
int main() {
```

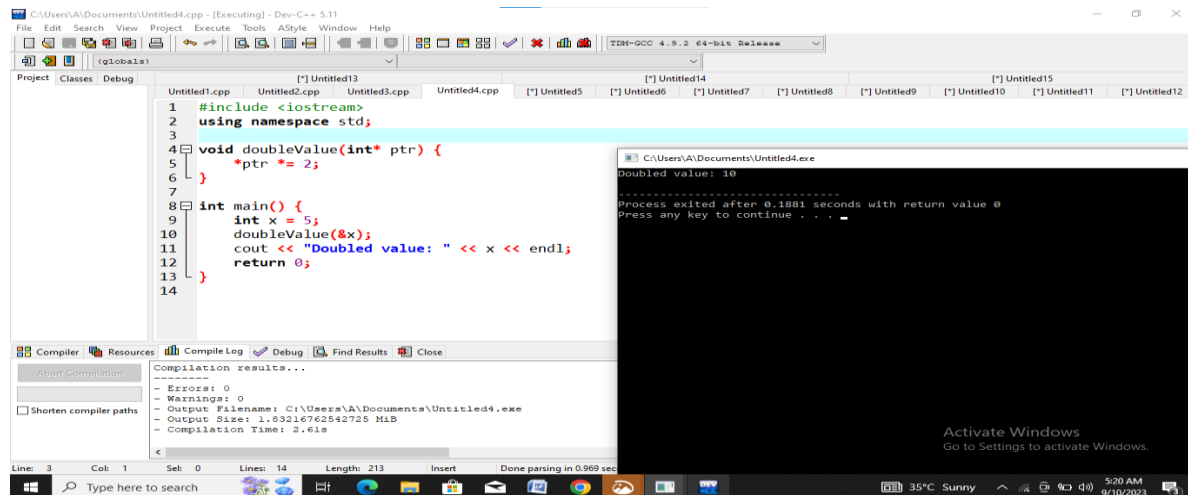
```
    int x = 5;
```

```
    doubleValue(&x);
```

```
    cout << "Doubled value: " << x << endl;
```

```
    return 0;
```

```
}
```



```
1 #include <iostream>
2 using namespace std;
3
4 void doubleValue(int* ptr) {
5     *ptr *= 2;
6 }
7
8 int main() {
9     int x = 5;
10    doubleValue(&x);
11    cout << "Doubled value: " << x << endl;
12    return 0;
13 }
14
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\A\Documents\Untitled4.exe
- Output Size: 1.83216762542725 MiB
- Compilation Time: 2.61s

Doubled value: 10

Process exited after 0.1881 seconds with return value 0

Press any key to continue . . .

5. Dynamic Memory Allocation:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int* ptr = new int;
```

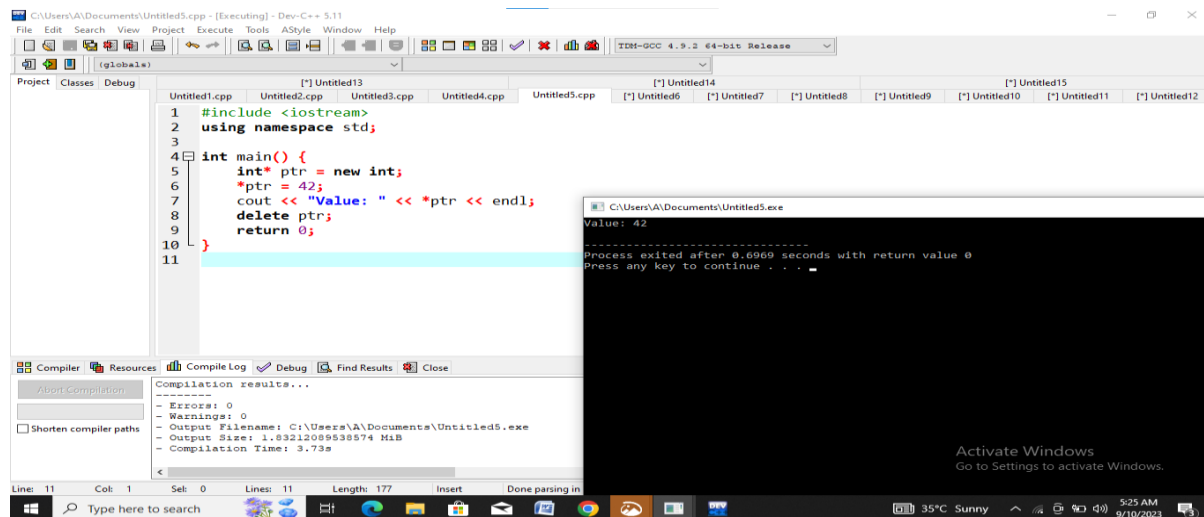
```
    *ptr = 42;
```

```
    cout << "Value: " << *ptr << endl;
```

```
    delete ptr;
```

```
    return 0;
```

```
}
```



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int* ptr = new int;
6     *ptr = 42;
7     cout << "Value: " << *ptr << endl;
8     delete ptr;
9     return 0;
10 }
11
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\A\Documents\Untitled5.exe
- Output Size: 1.83212089538574 MiB
- Compilation Time: 3.73s

Value: 42

Process exited after 0.6969 seconds with return value 0

Press any key to continue . . .

6. Pointer to Array:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

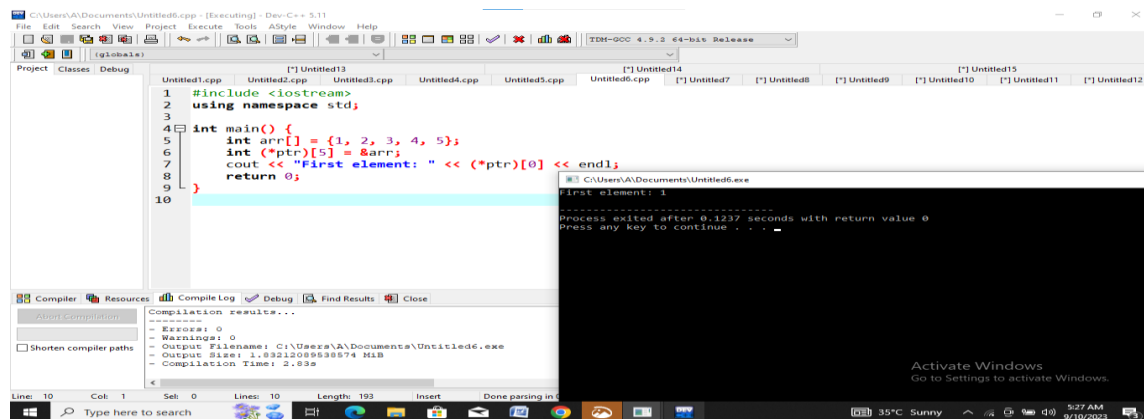
```
    int arr[] = {1, 2, 3, 4, 5};
```

```
    int (*ptr)[5] = &arr;
```

```
    cout << "First element: " << (*ptr)[0] << endl;
```

```
    return 0;
```

```
}
```



7. Pointer to Pointer:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

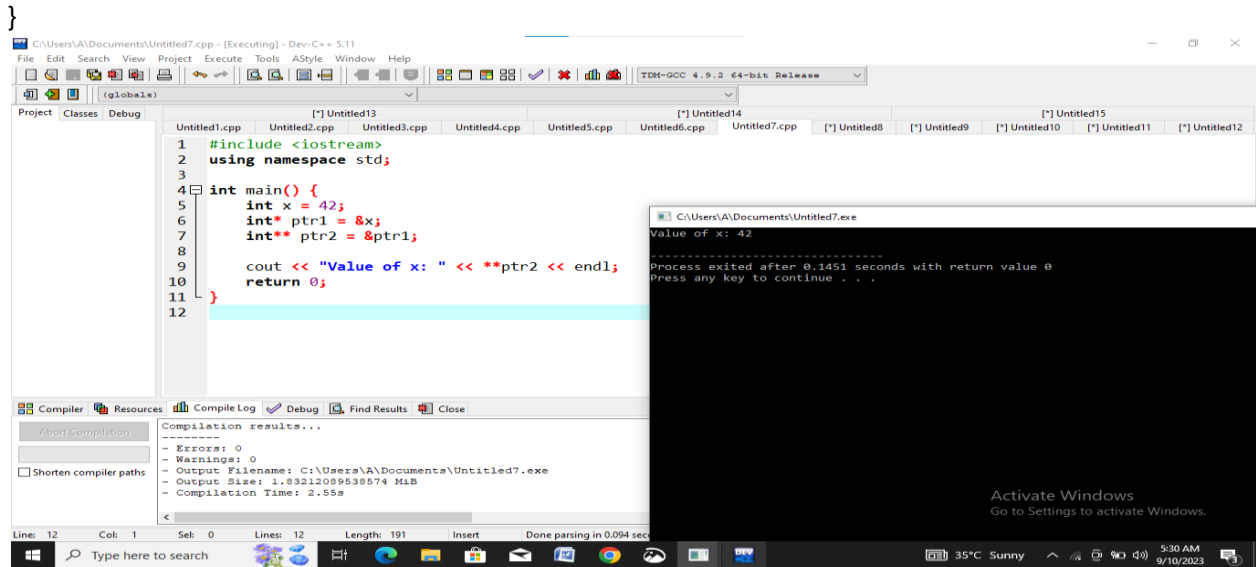
```
    int x = 42;
```

```
    int* ptr1 = &x;
```

```
    int** ptr2 = &ptr1;
```

```
    cout << "Value of x: " << **ptr2 << endl;
```

```
    return 0;
```



8. Pointers with Strings:

```
#include <iostream>
```

```
using namespace std;
```

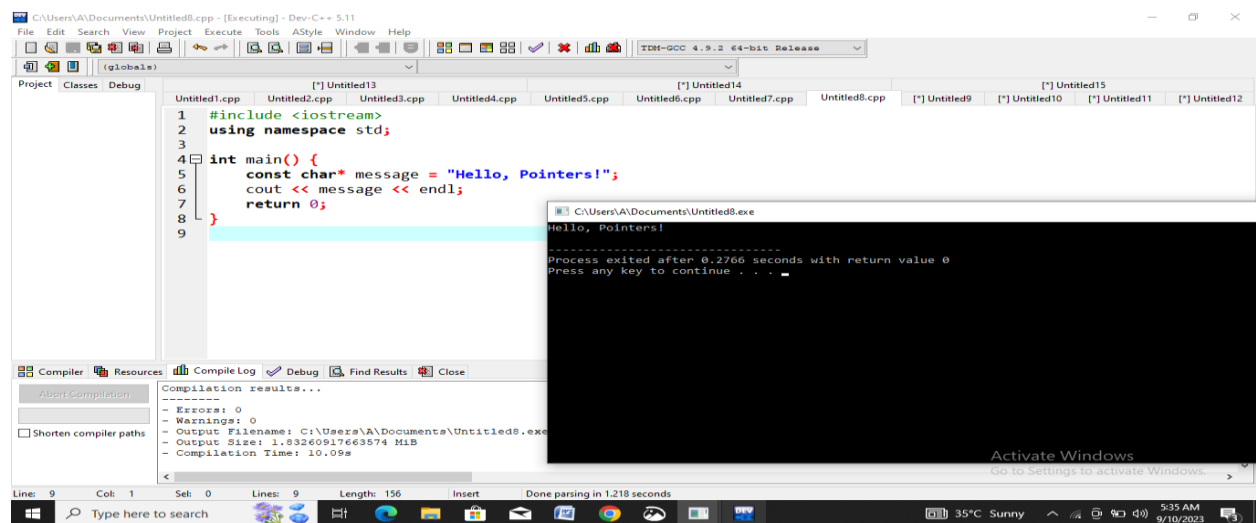
```
int main() {
```

```
    const char* message = "Hello, Pointers!";
```

```
    cout << message << endl;
```

```
    return 0;
```

```
}
```



9. Pointers to Structs:

```
#include <iostream>
```

```
using namespace std;
```

```
struct Point {
```

```
    int x;
```

```
    int y;
```

```
};
```

```
int main() {
```

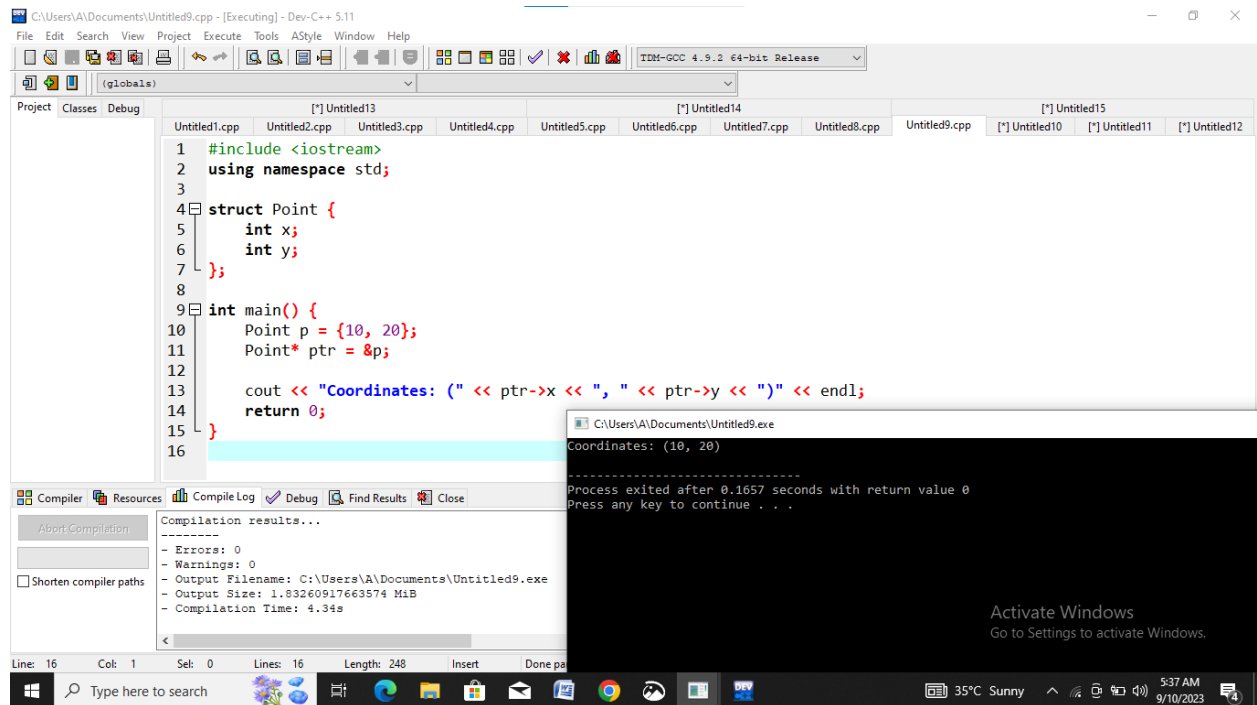
```
    Point p = {10, 20};
```

```
    Point* ptr = &p;
```

```
    cout << "Coordinates: (" << ptr->x << ", " << ptr->y << ")" << endl;
```

```
    return 0;
```

```
}
```



10. Null Pointers:

```
#include <iostream>
```



```

using namespace std;

int main() {

    int* ptr = NULL; // Creating a null pointer

    if (ptr == NULL) {

        cout << "Pointer is null." << endl;

    } else {

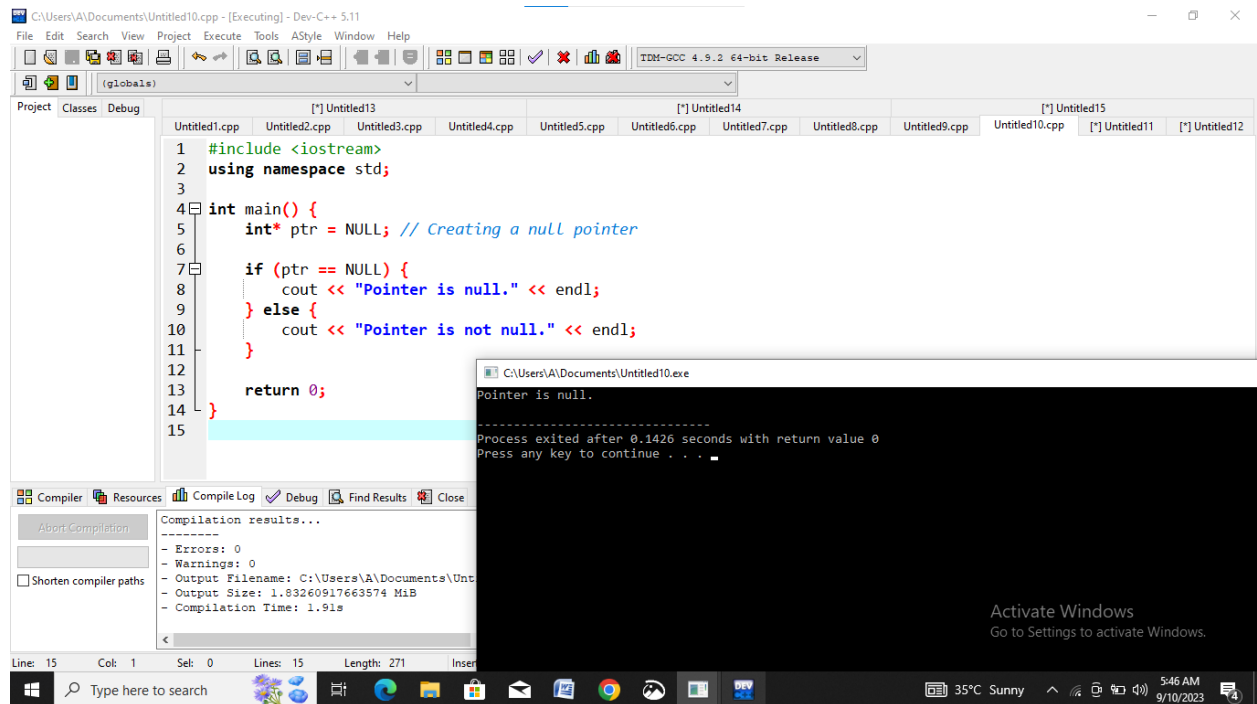
        cout << "Pointer is not null." << endl;

    }

    return 0;

}

```



11. Pointers to Constants:

```

#include <iostream>

using namespace std;

int main() {

    const int x = 42;

    const int* ptr = &x;

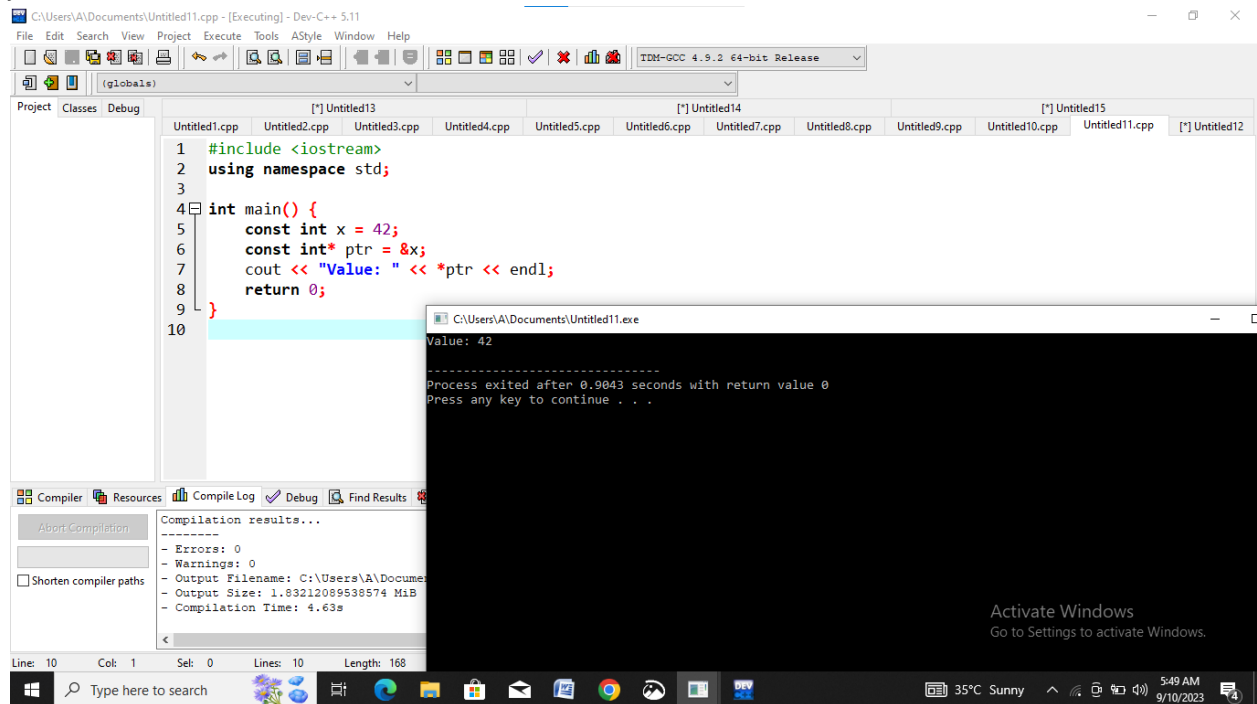
```

```

cout << "Value: " << *ptr << endl;

return 0;
}

```



12. Constant Pointers:

```

#include <iostream>

using namespace std;

int main() {

    int x = 42;

    int* const ptr = &x;

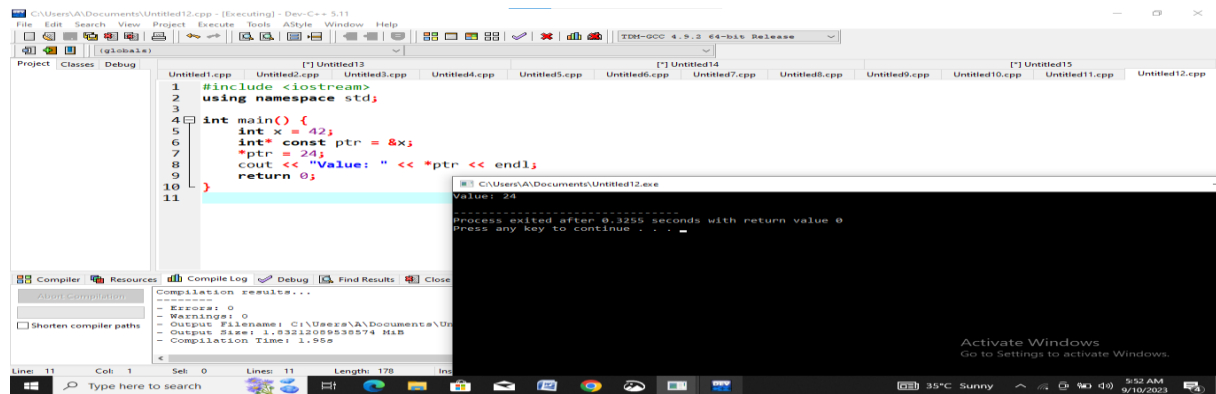
    *ptr = 24;

    cout << "Value: " << *ptr << endl;

    return 0;

}

```



13. Pointer to Constant Data:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int x = 42;
```

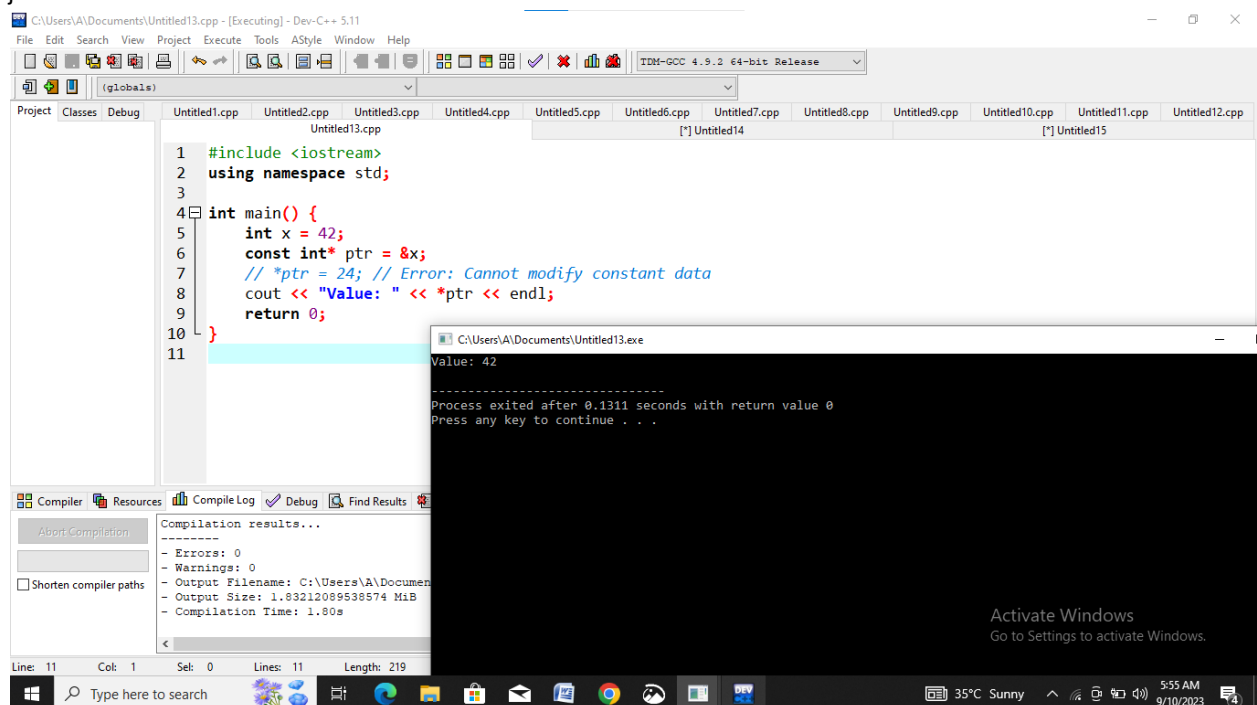
```
    const int* ptr = &x;
```

```
    // *ptr = 24; // Error: Cannot modify constant data
```

```
    cout << "Value: " << *ptr << endl;
```

```
    return 0;
```

```
}
```



14. Array of Pointers:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

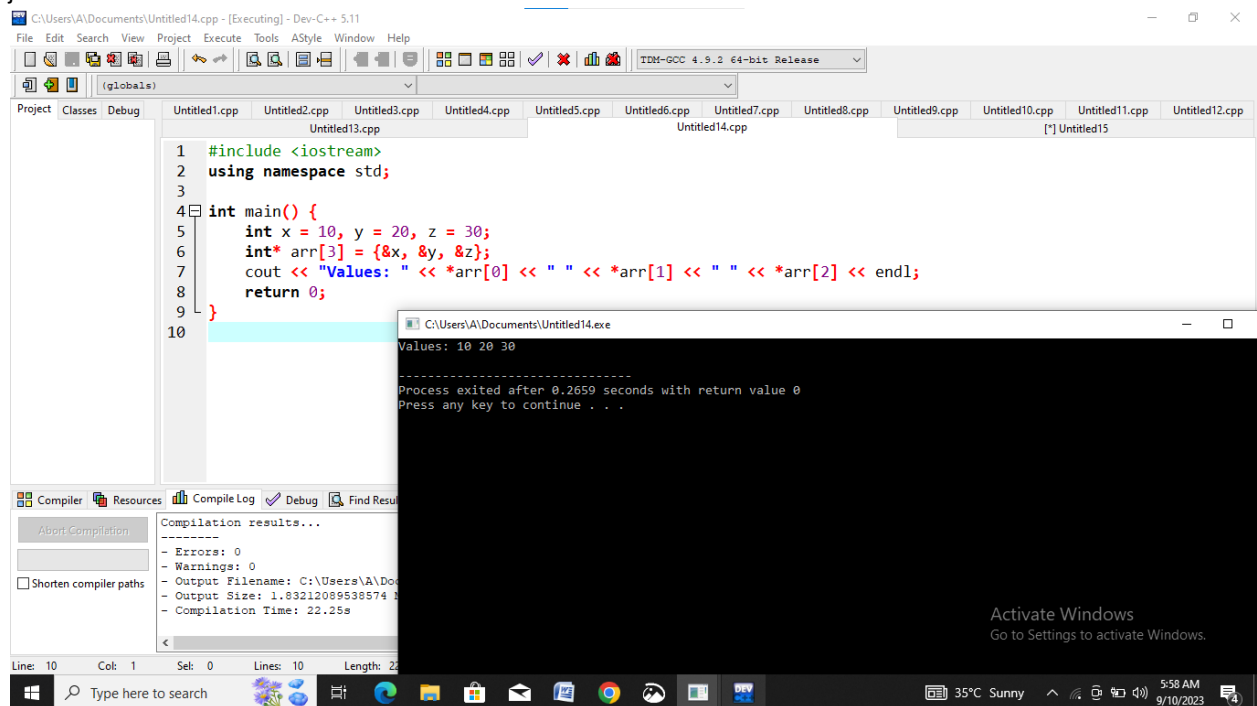
```
    int x = 10, y = 20, z = 30;
```

```
    int* arr[3] = {&x, &y, &z};
```

```
    cout << "Values: " << *arr[0] << " " << *arr[1] << " " << *arr[2] << endl;
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with the following code in the editor:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int x = 10, y = 20, z = 30;
6     int* arr[3] = {&x, &y, &z};
7     cout << "Values: " << *arr[0] << " " << *arr[1] << " " << *arr[2] << endl;
8     return 0;
9 }
10
```

The output window shows the execution result:

```
Values: 10 20 30
-----
Process exited after 0.2659 seconds with return value 0
Press any key to continue . . .
```

The compiler window shows the following compilation results:

```
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\A\Documents\Untitled14.exe
- Output Size: 1.83212089538574 KB
- Compilation Time: 22.25s
```

15. Pointers to Functions:

```
#include <iostream>
```

```
using namespace std;
```

```
int add(int a, int b) {
```

```
    return a + b;
```

```
}
```

```
int main() {
```

```
int (*ptr)(int, int) = add;

cout << "Sum: " << ptr(5, 7) << endl;

return 0;
```

```
}
```

