



Assignment 01

Name:	Nimra Javeed
Reg No:	Sp22-Bcs-065
Subject:	Data Structure & Algorithm Lab
Submission Date:	10/Sep/2023
Submitted To:	Ma'am Yasmeen Jana

**Cosat University Islamabad Vehari
Campus**

Question 01: What a process to Create GitHub account?

1. Open a Web Browser:

Open your preferred web browser and go to the GitHub website at <https://github.com>.

2. Sign UP:

On the GitHub homepage, you will see a "Sign up" button in the upper right corner. Click on it.

3. Create GitHub Account:

You will be directed to a sign-up page. Here, you will need to provide the following information:

Username:

Choose a unique username for your GitHub account. This will also be a part of your GitHub profile URL (e.g., <https://github.com/your-username>).

Email Address:

Enter your email address. This will be used for account notifications and password resets.

Password:

Create a strong password for your GitHub account.

4. Verify Your Email Address:

After you've filled in the required information, GitHub will send you a verification email to the address you provided. Open your email inbox and click the verification link in the email to confirm your email address.

5. Choose a Plan:

GitHub offers both free and paid plans. For most users, the free plan (GitHub Free) is sufficient. Click on the "Choose Free" button to continue.

6. Welcome to GitHub:

Once you are completed all the steps, you'll be directed to your GitHub dashboard, and you'll officially have a GitHub account.

Question 02: Write any 15 programs that will explain the concepts of pointer.

1: Pointer declaration and initialization with user input.

```
#include <iostream>

int main() {

    int a;

    int *ptr;

    std::cout << "Enter an integer: ";

    std::cin >> a;

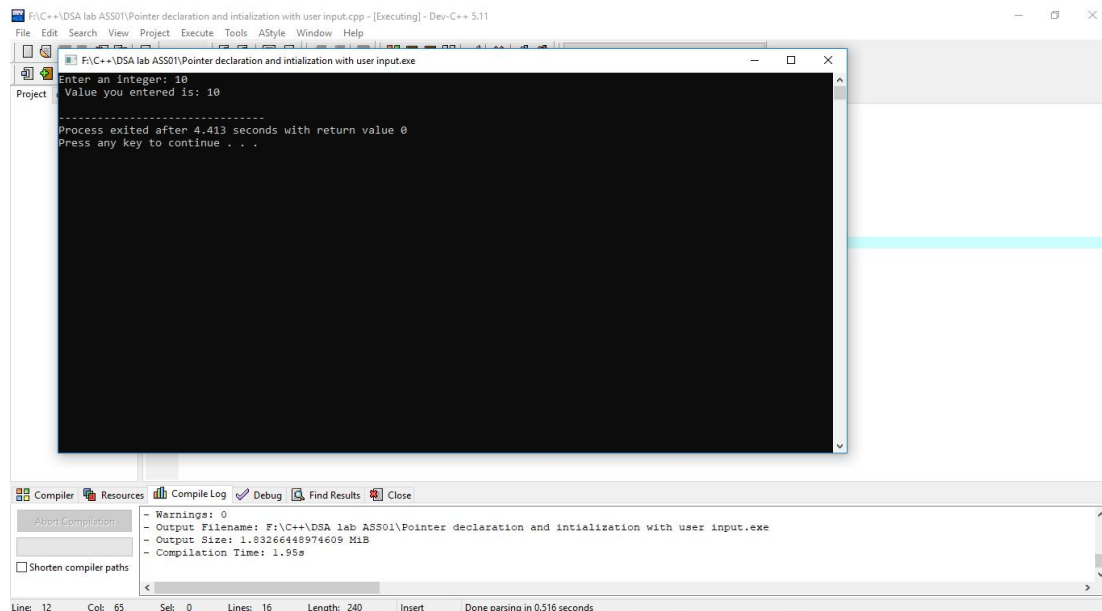
    ptr = &a;

    std::cout << " Value you entered is: " << *ptr << std::endl;

    return 0;

}
```

Output



02: Add two numbers.

```
#include <iostream>
```

```
int main() {
```

```
    int num1, num2;
```

```
    int *ptr;
```

```
    std::cout << "Enter the first integer: ";
```

```
    std::cin >> num1;
```

```
    std::cout << "Enter the second integer: ";
```

```
    std::cin >> num2;
```

```
    ptr = &num1;
```

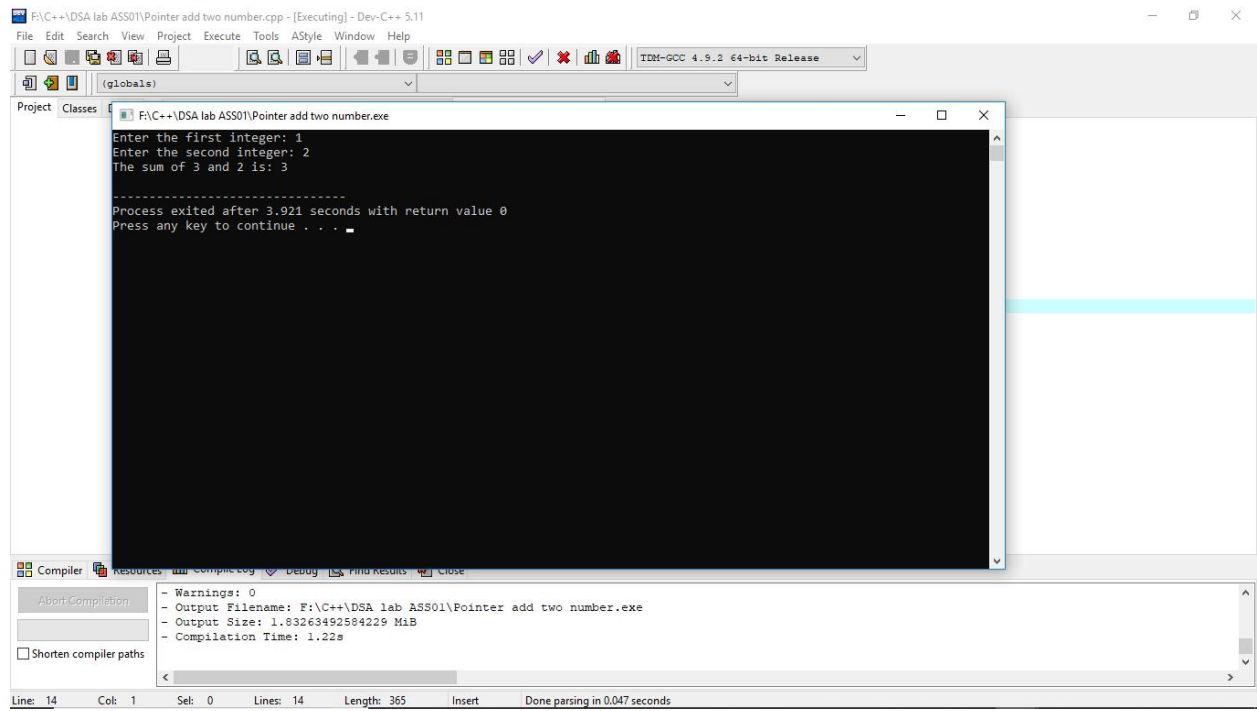
```
    *ptr += num2;
```

```
    std::cout << "The sum of " << num1 << " and " << num2 << " is: " << *ptr << std::endl;
```

```
    return 0;
```

```
}
```

Output



```
F:\C++\DSA lab AS501\Pointer add two number.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes
F:\C++\DSA lab AS501\Pointer add two number.exe
Enter the first integer: 1
Enter the second integer: 2
The sum of 3 and 2 is: 3
-----
Process exited after 3.921 seconds with return value 0
Press any key to continue . . .
Compiler Resources Compile Log Debug Find Results Close
- Warnings: 0
- Output Filename: F:\C++\DSA lab AS501\Pointer add two number.exe
- Output Size: 1.83263492584229 MiB
- Compilation Time: 1.22s
Shorten compiler paths
Line: 14 Col: 1 Sel: 0 Lines: 14 Length: 365 Insert Done parsing in 0.047 seconds
```

03: Pointer Arithmetic.

```
#include <iostream>
```

```
int main() {
```

```
    int numbers[] = {10, 20, 30, 40, 50};
```

```
    int *ptr = numbers;
```

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

```
        std::cout << " A " << i << ": " << *ptr << std::endl;
```

```

        ptr++;

    }

    return 0;

}

```

Output:

```

F:\C++\DSA lab ASS01\Pointer Arithmetic.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
F:\C++\DSA lab ASS01\Pointer Arithmetic.exe
A 0: 10
A 1: 20
A 2: 30
A 3: 40
A 4: 50
-----
Process exited after 0.1422 seconds with return value 0
Press any key to continue . . .
Compiler Resources Compile Log Debug Find Results Close
About Completion
Shorten compiler paths
- Warnings: 0
- Output Filename: F:\C++\DSA lab ASS01\Pointer Arithmetic.exe
- Output Size: 1.83214282989502 MiB
- Compilation Time: 0.92s
Line: 12 Col: 1 Sel: 0 Lines: 12 Length: 247 Insert Done parsing in 0.015 seconds

```

04: Accessing the value.

```
#include <iostream>
```

```
int main() {
```

```
    int number=10000;
```

```
    int *ptr=&number;
```

```

int value=*ptr;

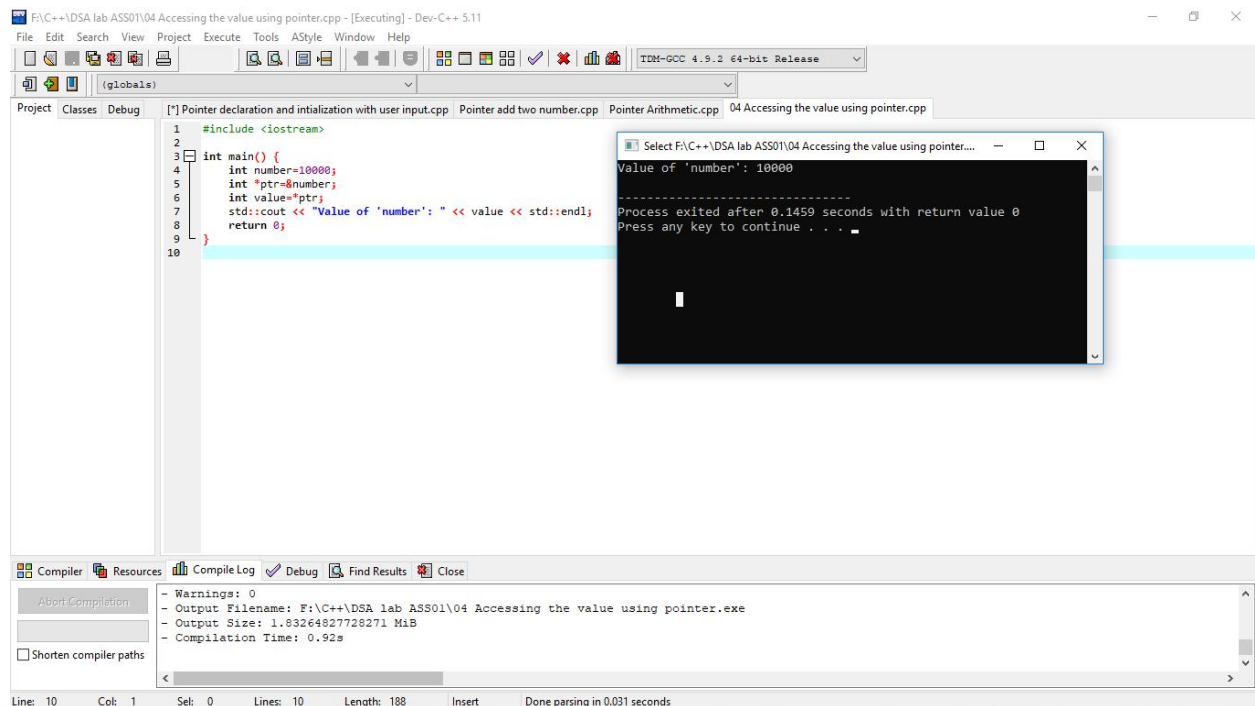
std::cout << "Value of 'number': " << value << std::endl;

return 0;

}

```

Output:



05: Function Pointer.

```
#include <iostream>
```

```

int add(int a, int b) {

    return a + b;

}

```

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

```
    return a - b;
```

```
}
```

```
int main() {
```

```
    int (*operation)(int, int);
```

```
    operation = add;
```

```
    std::cout << "Addition: " << operation(5, 3) << std::endl;
```

```
    operation = subtract;
```

```
    std::cout << "Subtraction: " << operation(5, 3) << std::endl;
```

```
    return 0;
```

```
}
```


Output:

The screenshot shows a C++ IDE with a project named "F:\C++\DSA lab ASS01\06 Function Pointer.cpp". The code defines two functions, `add` and `subtract`, and a `main` function that uses function pointers to call them. The output window shows the results of these calls: "Addition: 8" and "Subtraction: 2". The compiler log at the bottom indicates that the program compiled successfully with no warnings.

```
#include <iostream>

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

int subtract(int a, int b) {
    return a - b;
}

int main() {
    int (*operation)(int, int);

    operation = add;
    std::cout << "Addition: " << operation(5, 3) << std::endl;

    operation = subtract;
    std::cout << "Subtraction: " << operation(5, 3) << std::endl;

    return 0;
}
```

Output:

```
Addition: 8
Subtraction: 2

Process exited after 0.1506 seconds with return value 0
Press any key to continue . . .
```

Compiler Log:

```
- Warnings: 0
- Output Filename: F:\C++\DSA lab ASS01\06 Function Pointer.exe
- Output Size: 1.83268356323242 MiB
- Compilation Time: 1.30s
```

06: Pointer to pointer.

```
#include <iostream>
```

```
int main() {

    int number = 12;

    int *ptr1 = &number;

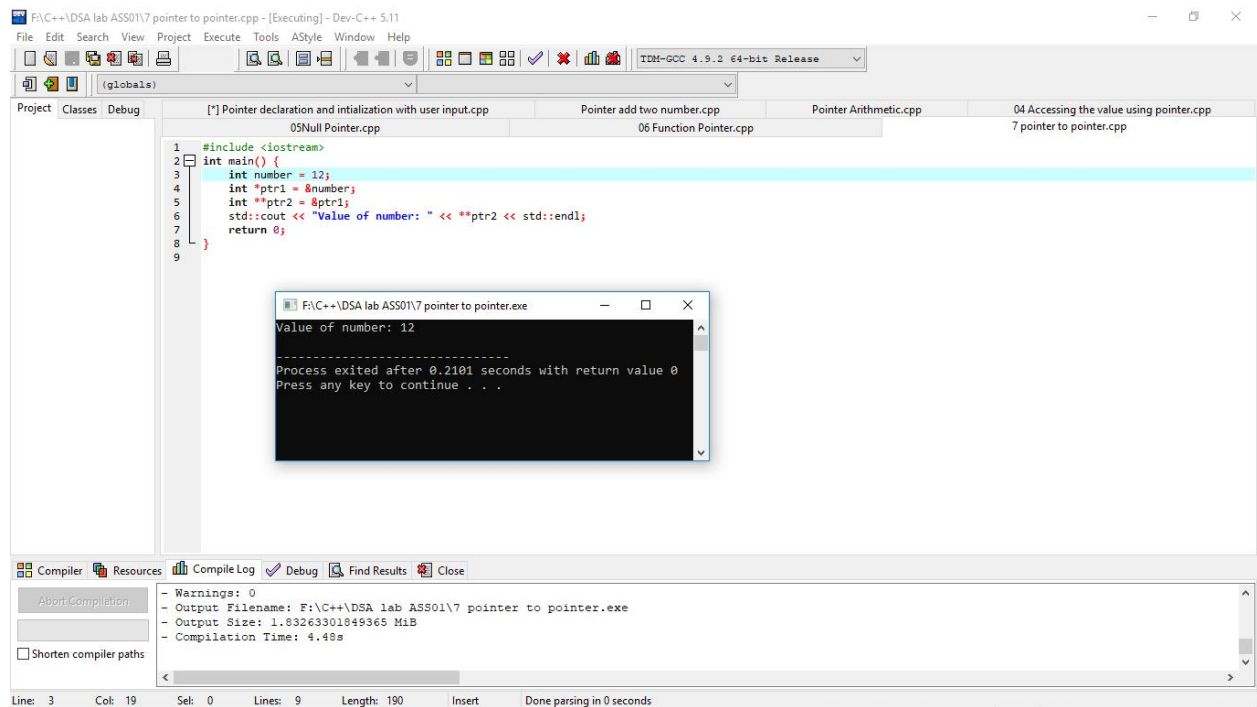
    int **ptr2 = &ptr1;

    std::cout << "Value of number: " << **ptr2 << std::endl;

    return 0;

}
```

Output:



07: Array of pointer.

```
#include <iostream>
```

```
int main() {
```

```
    int a = 5, b = 10, c = 15;
```

```
    int *ptrArr[3] = {&a, &b, &c};
```

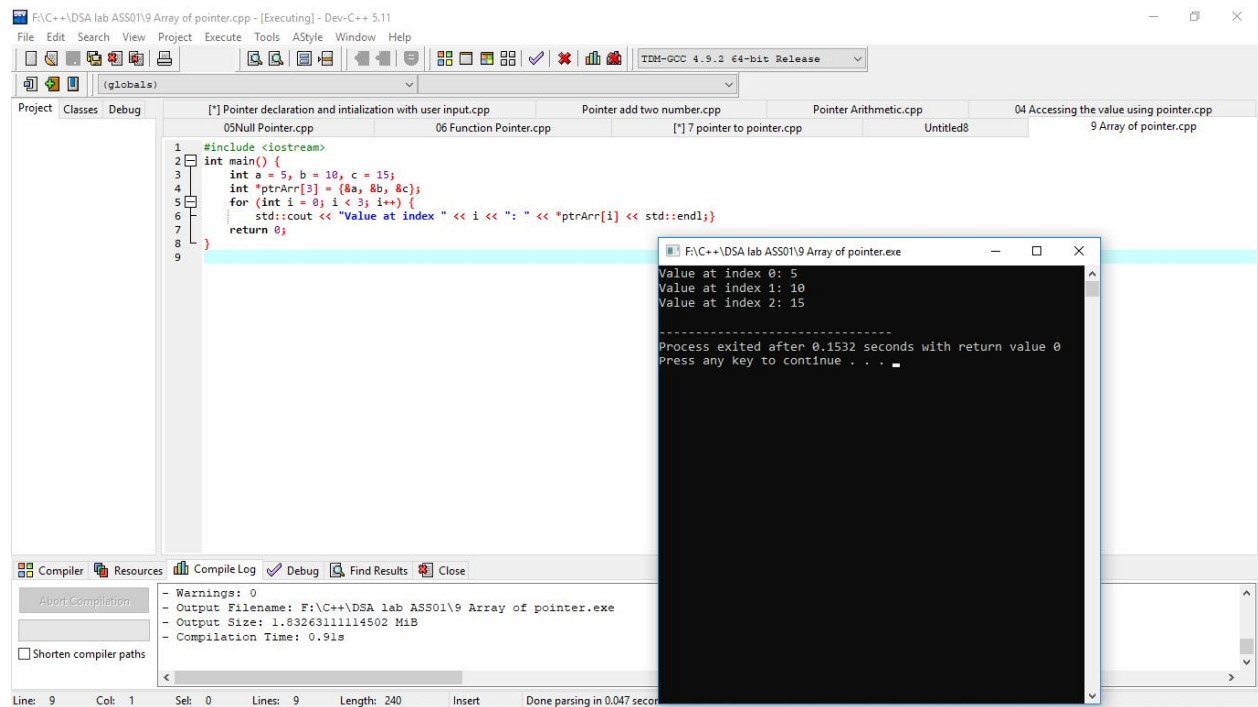
```
    for (int i = 0; i < 3; i++) {
```

```
        std::cout << "Value at index " << i << ": " << *ptrArr[i] << std::endl;
```

```
    return 0;
```

```
}
```

Output:



08: Pointer using Array.

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    float arr[3];
```

```
    float *ptr;
```

```
    cout << "Displaying address using arrays: " << endl;
```

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

```
{
```

```

        cout << "&arr[" << i << "]" = " << &arr[i] << endl;

    }

    ptr = arr;

    cout<<"\nDisplaying address using pointers: "<< endl;

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

    {

        cout << "ptr + " << i << " = "<< ptr + i << endl;

    }

    return 0;

}

```

Output:

The screenshot shows the Dev-C++ IDE with the following components:

- Source Code Window:** Displays the C++ program '10 pointer using array.cpp'. The code includes `<iostream>`, uses the `std` namespace, and defines a `main` function. Inside `main`, a float array `arr` of size 3 is declared, and a pointer `ptr` is initialized to point to `arr`. The program prints the addresses of `arr` elements using both `&arr[i]` and `ptr + i` notation.
- Output Window:** Shows the execution output. It displays the addresses of `&arr[0]`, `&arr[1]`, and `&arr[2]`, followed by the addresses calculated using pointer arithmetic: `ptr + 0`, `ptr + 1`, and `ptr + 2`. The output confirms that the pointer arithmetic correctly calculates the addresses of the array elements.
- Compiler Window:** Shows the compilation results, indicating 0 warnings and 0 errors. The output filename is `F:\C++\DSA lab AS501\10 pointer using array.exe`, and the compilation time is 3.16s.

09:

```
#include <iostream>

int main() {

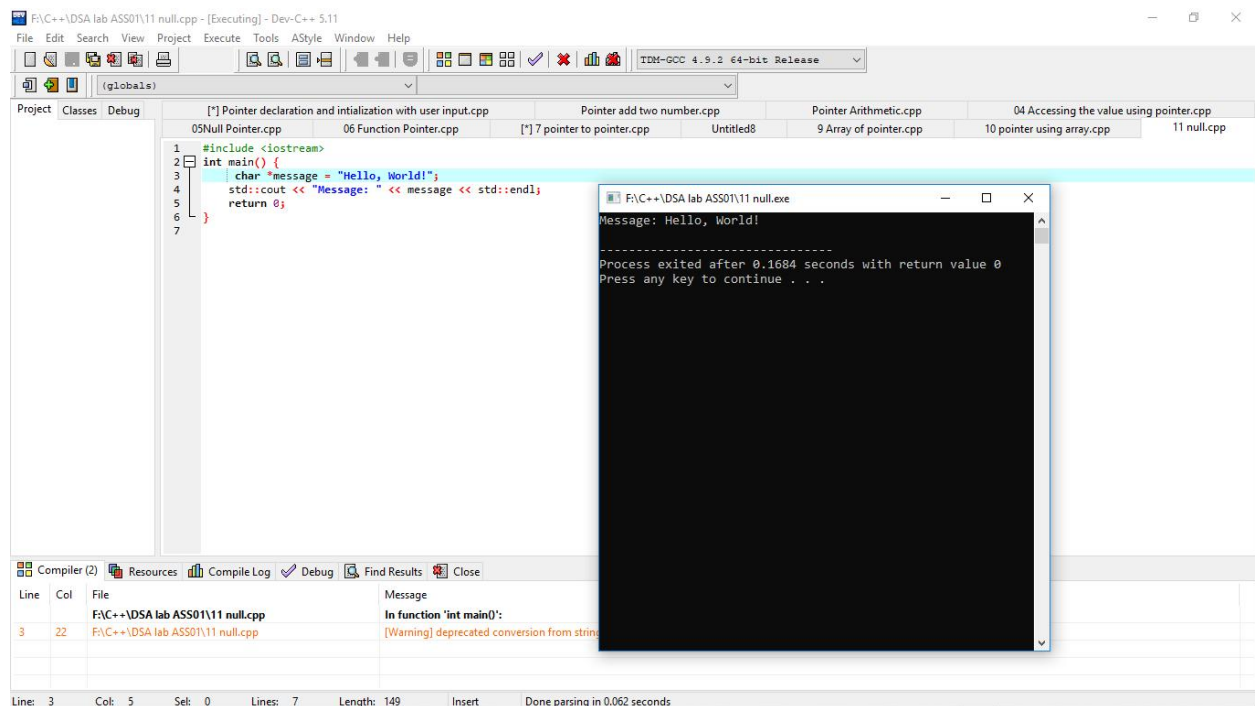
    char *message = "Hello, World!";

    std::cout << "Message: " << message << std::endl;

    return 0;

}
```

Output:



10: Pointer to Object.

```
#include <iostream>

class MyClass {

public:

    int data;
```

```

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

};

int main() {

    MyClass obj(42);

    MyClass *ptr = &obj;

    std::cout << "Value of data: " << ptr->data << std::endl;

    return 0;

}

```

Output:

The screenshot displays the Dev-C++ IDE with the following code in the editor:

```

1 #include <iostream>
2 class MyClass {
3 public:
4     int data;
5     MyClass(int val) : data(val) {}
6 }
7
8 int main() {
9     MyClass obj(42);
10    MyClass *ptr = &obj;
11
12    std::cout << "Value of data: " << ptr->data << std::endl;
13    return 0;
14 }

```

The output window shows the following text:

```

Value of data: 42
-----
Process exited after 0.1941 seconds with return value 0
Press any key to continue . . .

```

The status bar at the bottom indicates: Line: 1, Col: 20, Sel: 0, Lines: 14, Length: 250, Insert, Done parsing in 0.016 seconds.

11: Pointer to constant data.

```
#include <iostream>
```

```

int main() {

    const int a = 2;

    const int *ptr = &a;

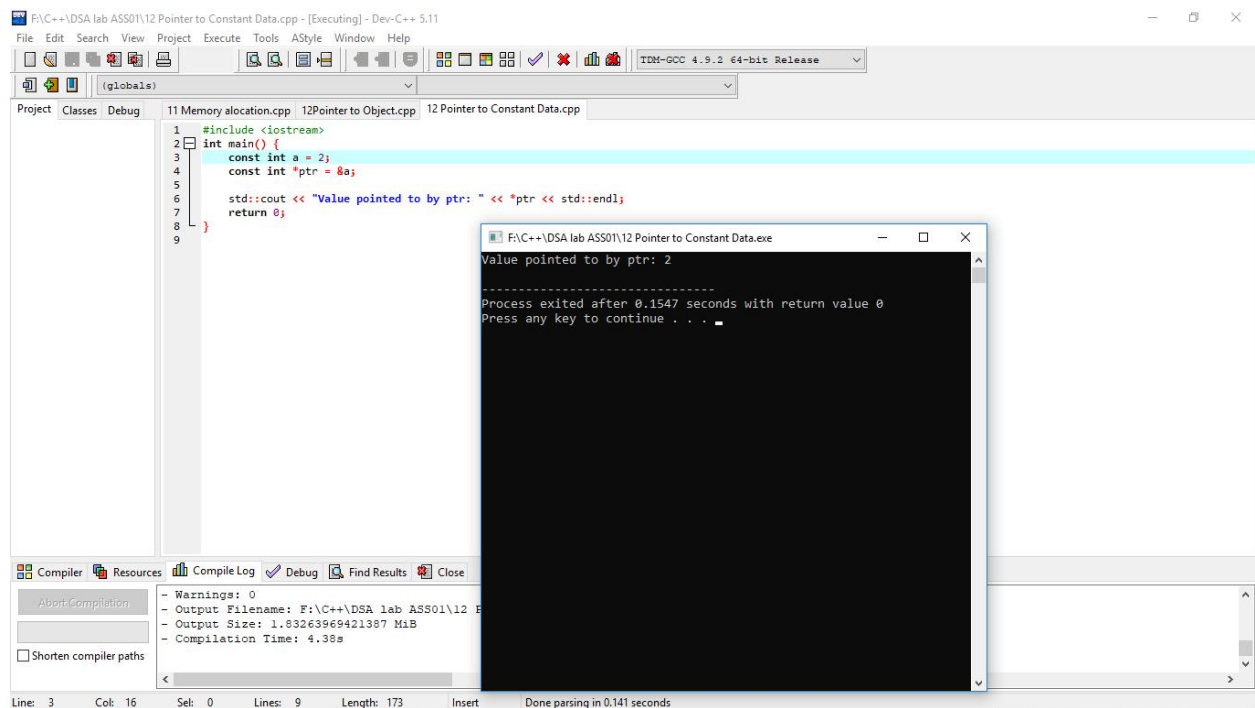

    std::cout << "Value pointed to by ptr: " << *ptr << std::endl;

    return 0;

}

```

Output:



12: Pointer to constant pointer.

```
#include <iostream>
```

```
int main() {
```

```
    int x = 10;
```

```

int *const ptr = &x;

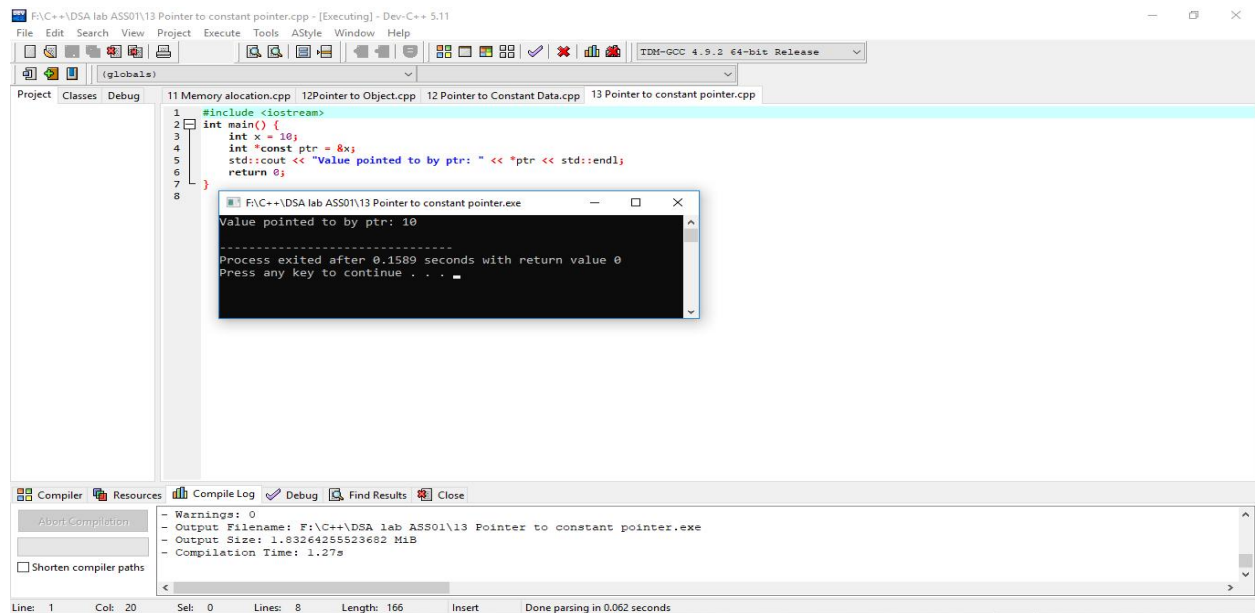
std::cout << "Value pointed to by ptr: " << *ptr << std::endl;

return 0;

}

```

Output:



13: Pointer to with different function.

```
#include <iostream>
```

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

```
    return a + b;
```

```
}
```

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



```

        return a - b;

    }

int main() {

    int (*ptr)(int, int);

    ptr = &add;

    std::cout << "Result of addition: " << (*ptr)(4, 5) << std::endl;

    ptr = &subtract;

    std::cout << "Result of subtraction: " << (*ptr)(9, 3) << std::endl;

    return 0;

}

```

Output:

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

```

1 #include <iostream>
2 int add(int a, int b) {
3     return a + b;
4 }
5
6 int subtract(int a, int b) {
7     return a - b;
8 }
9
10 int main() {
11     int (*ptr)(int, int);
12     ptr = &add;
13     std::cout << "Result of addition: " << (*ptr)(4, 5) << std::endl;
14     ptr = &subtract;
15     std::cout << "Result of subtraction: " << (*ptr)(9, 3) << std::endl;
16     return 0;
17 }

```

The output window displays the following text:

```

Result of addition: 9
Result of subtraction: 6
Press any key to continue . . .

```

The status bar at the bottom indicates the file path: F:\C++\DSA lab ASS01\13 Pointer to with different function.exe. The compilation time is 0.41s.

14: Multiply pointer.

```
#include <iostream>
```

```
int main() {
```

```

int num1 = 5;

int num2 = 7;

int result = 0;

int *ptr1 = &num1;

int *ptr2 = &num2;

result = (*ptr1) * (*ptr2);

std::cout << "Multiplication result: " << result << std::endl;

return 0;

}

```

Output:

The screenshot shows a C++ IDE with the following components:

- Code Editor:** Displays the source code for 'multiply pointer.cpp'. The code defines two integers, num1 (5) and num2 (7), and pointers ptr1 and ptr2 that point to their memory addresses. It calculates the product of the values pointed to by ptr1 and ptr2, storing the result in 'result', and prints it using std::cout.
- Compiler Output:** Shows the compilation process. It indicates that there were 0 warnings, the output filename is 'F:\C++\DSA lab ASS01\14 multiply pointer.exe', the output size is 1.83263206491934 MiB, and the compilation time was 2.63s.
- Output Window:** Displays the runtime output of the program, which is 'Multiplication result: 35'. It also shows the process exiting after 0.3056 seconds with a return value of 0.

15: Divide function.

```
#include <iostream>
```

```

int main() {

    int num1 = 15;

    int num2 = 7;

    int result = 0;

    int *ptr1 = &num1;

    int *ptr2 = &num2;

    result = (*ptr1) / (*ptr2);

    std::cout << "Divide result: " << result << std::endl;

    return 0;

}

```

Output:

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

```

1 #include <iostream>
2
3 int main() {
4     int num1 = 15;
5     int num2 = 7;
6     int result = 0;
7     int *ptr1 = &num1;
8     int *ptr2 = &num2;
9     result = (*ptr1) / (*ptr2);
10    std::cout << "Divide result: " << result << std::endl;
11    return 0;
12 }
13

```

The output window displays the following text:

```

Divide result: 2

Process exited after 0.3386 seconds with return value 0
Press any key to continue . . .

```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, a project explorer, and a compiler output window at the bottom showing warnings and compilation details.