

## Program 1:

### Introduction to pointers

```
#include <iostream>
using namespace std;

int main() {
    // Program 1: Introduction to Pointers
    int num = 42;
    int* ptr = &num;

    cout << "Program 1: Introduction to Pointers" << endl;
    cout << "Value of num: " << num << endl;
    cout << "Address of num: " << &num << endl;
    cout << "Value of ptr: " << *ptr << endl;
    cout << "Address stored in ptr: " << ptr << endl;
    cout << endl;
}
```

### Output:

```
Program 2: Pointer Arithmetic
Element at arrPtr: 10
Next element: 20
Previous element: 0

-----
Process exited after 0.2129 seconds with return value 0
Press any key to continue . . .
```

## Program2:

### Pointers Arithmetic

```
#include <iostream>
using namespace std;

int main() {
    // Program 2: Pointer Arithmetic
    int arr[5] = {10, 20, 30, 40, 50};
    int* arrPtr = arr;

    cout << "Program 2: Pointer Arithmetic" << endl;
    cout << "Element at arrPtr: " << *arrPtr << endl;
    cout << "Next element: " << *(arrPtr + 1) << endl;
    cout << "Previous element: " << *(arrPtr - 1) << endl;
    cout << endl;
}
```

### Output:

```

Program 2: Pointer Arithmetic
Element at arrPtr: 10
Next element: 20
Previous element: 0

-----
Process exited after 0.2129 seconds with return value 0
Press any key to continue . . .

```

### Program 3: Dynamic Memory Allocation

```

#include <iostream>
using namespace std;

int main() {
    // Program 5: Dynamic Memory Allocation
    int* dynamicArray = new int[3];
    dynamicArray[0] = 100;
    dynamicArray[1] = 200;
    dynamicArray[2] = 300;

    cout << "Program 5: Dynamic Memory Allocation" << endl;
    for (int i = 0; i < 3; i++) {
        cout << "Element " << i << ": " << dynamicArray[i] << endl;
    }
    delete[] dynamicArray;
    cout << endl;
}

```

### Output

```

C:\Users\sahib\OneDrive\Documents\Dynamic Memory Allocation.exe
Program 5: Dynamic Memory Allocation
Element 0: 100
Element 1: 200
Element 2: 300

-----
Process exited after 0.1943 seconds with return value 0
Press any key to continue . . .

```

### Program 7

```

#include <iostream>
using namespace std;

int main(){
    // Program 7: Pointer to Pointer
    int value = 42;
    int* ptr1 = &value;
    int** ptr2 = &ptr1;

    cout << "Program 7: Pointer to Pointer" << endl;
    cout << "Value of value: " << value << endl;
    cout << "Value of ptr1: " << *ptr1 << endl;
    cout << "Value of ptr2: " << **ptr2 << endl;
    cout << endl;

}

```

## Output:

```

C:\Users\sahib\OneDrive\Documents\pointers to function.exe
Program 7: Pointer to Pointer
Value of value: 42
Value of ptr1: 42
Value of ptr2: 42

-----
Process exited after 0.2178 seconds with return value 0
Press any key to continue . . .

```

## Program 8

```

#include <iostream>
using namespace std;

int main(){
    // Program 8: Pointers to Objects
    class MyClass {
    public:
        void display() {
            cout << "Hello from MyClass!" << endl;
        }
    };
    MyClass obj;
    MyClass* objPtr = &obj;

    cout << "Program 8: Pointers to Objects" << endl;
    objPtr->display();
    cout << endl;

}

```

## Output:

```
C:\Users\sahib\OneDrive\Documents\pointers to object.exe
Program 8: Pointers to Objects
Hello from MyClass!

-----
Process exited after 0.02412 seconds with return value 0
Press any key to continue . . .
```

## Program 9:

```
#include <iostream>
using namespace std;

int main(){
    // Program 9: Pointers and Arrays
    int array[] = {10, 20, 30};
    int* p = array;

    cout << "Program 9: Pointers and Arrays" << endl;
    cout << "Array Element 0: " << *p << endl;
    cout << "Array Element 1: " << *(p + 1) << endl;
    cout << "Array Element 2: " << *(p + 2) << endl;
    cout << endl;

}
```

## Output:

```
Program 9: Pointers and Arrays
Array Element 0: 10
Array Element 1: 20
Array Element 2: 30

-----
Process exited after 0.2068 seconds with return value 0
Press any key to continue . . .
```

## Program 10:

```

#include <iostream>
using namespace std;

int main(){
    // Program 10: Dynamic Memory Allocation for Objects
    class Person {
    public:
        string name;
        int age;
    };
    Person* personPtr = new Person;
    personPtr->name = "Alice";
    personPtr->age = 25;

    cout << "Program 10: Dynamic Memory Allocation for Objects" << endl;
    cout << "Name: " << personPtr->name << ", Age: " << personPtr->age << endl;
    delete personPtr;
    cout << endl;

}

```

### Output:

```

Program 10: Dynamic Memory Allocation for Objects
Name: Alice, Age: 25

-----
Process exited after 0.2062 seconds with return value 0
Press any key to continue . . .

```

### Program 11:

```

#include <iostream>
using namespace std;

int main(){
    // Program 11: Pointers to Constant Data
    const int constValue = 100;
    const int* constPtr = &constValue;

    cout << "Program 11: Pointers to Constant Data" << endl;
    cout << "Value of constValue: " << constValue << endl;
    cout << "Value of constPtr: " << *constPtr << endl;
    cout << endl;

}

```

### Output:



```
Program 11: Pointers to Constant Data
Value of constValue: 100
Value of constPtr: 100

-----
Process exited after 0.2097 seconds with return value 0
Press any key to continue . . .
```

## Program 12:

```
#include <iostream>
using namespace std;

int main(){
    // Program 12: Pointers to Arrays with Dynamic Memory Allocation
    int size = 4;
    int* dynamicArr = new int[size];

    cout << "Program 12: Pointers to Arrays with Dynamic Memory Allocation" << endl;
    for (int i = 0; i < size; i++) {
        dynamicArr[i] = i * 10;
        cout << "Element " << i << ": " << dynamicArr[i] << endl;
    }
    delete[] dynamicArr;
    cout << endl;
}
```

## Output:

```
Program 12: Pointers to Arrays with Dynamic Memory Allocation
Element 0: 0
Element 1: 10
Element 2: 20
Element 3: 30

-----
Process exited after 0.1746 seconds with return value 0
Press any key to continue . . .
```

## Program 13

```

#include <iostream>

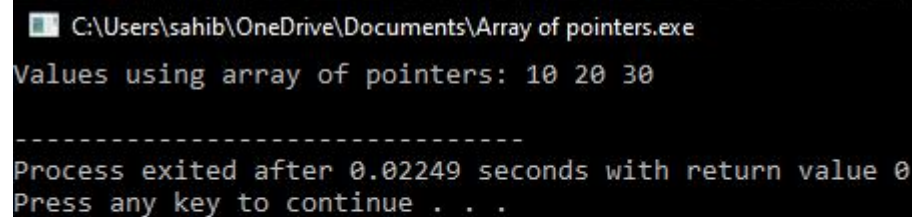
int main() {
    int x = 10, y = 20, z = 30;
    int *ptrArr[] = {&x, &y, &z}; // Array of pointers

    std::cout << "Values using array of pointers: ";
    for (int i = 0; i < 3; i++) {
        std::cout << *ptrArr[i] << " ";
    }
    std::cout << std::endl;

    return 0;
}

```

### Output:



C:\Users\sahib\OneDrive\Documents\Array of pointers.exe  
 Values using array of pointers: 10 20 30  
 -----  
 Process exited after 0.02249 seconds with return value 0  
 Press any key to continue . . .

### Program 14:

```

#include <iostream>
using namespace std;

int main() {
    int arr[] = {10, 20, 30};
    int *ptrArr[3];

    for (int i = 0; i < 3; i++) {
        ptrArr[i] = &arr[i];
        cout << "Value at ptrArr[" << i << "]: " << *ptrArr[i] << endl;
    }

    return 0;
}

```

### Output:

```
C:\Users\sahib\OneDrive\Documents\Array of pointers.exe
Value at ptrArr[0]: 10
Value at ptrArr[1]: 20
Value at ptrArr[2]: 30
-----
Process exited after 0.2344 seconds with return value 0
Press any key to continue . . .
```

### Program 15:

```
#include <iostream>
using namespace std;

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

int main() {
    int (*ptr)(int, int);
    ptr = add;

    cout << "Result of add(5, 7) using pointer to function: " << ptr(5, 7) << endl;

    return 0;
}
```

### Output:

```
C:\Users\sahib\OneDrive\Documents\pointers to function.exe
Result of add(5, 7) using pointer to function: 12
-----
Process exited after 0.2145 seconds with return value 0
Press any key to continue . . .
```