Contents

1 Pointers

A **pointer** is a variable that represents a memory address, often the address of another variable. Pointers “point” to locations in memory and allow indirect access to the values stored there.

1.1 Declaring and Defining Pointers

To declare a pointer, use an asterisk * and specify the type of data it points to:

```cpp
int* ptr; // declares a pointer to an int
```

To make a pointer “point” to a variable, use the address-of operator &:

```cpp
int* ptr;
int x {};
ptr = &x; // 'point' ptr to the address of x
```

1.2 Using Pointers

To access the value pointed to, we use the dereference operator *:

dereference.cpp

```cpp
#include <iostream>

int main()
{
    int x { 42 };
    int* ptr { &x };
    std::cout << "x = " << *ptr << std::endl;
}
```

Terminal

```
$ clang++ -std=c++23 dereference.cpp
$ ./a.out
x = 42
```

Since pointers provide direct memory access, they can also be used to modify variables indirectly:

increment.cpp

```cpp
#include <iostream>

int main()
{
    int x { 42 };
    int* ptr { &x };
    (*ptr)++;
    std::cout << "x = " << x << std::endl;
}
```

Terminal

```
$ clang++ -std=c++23 increment.cpp
$ ./a.out
x = 43
```