C++ Pointers

A pointer is a C++ data type. The value that a pointer can hold is a memory location. We say that a pointer variable *points* to the memory location. There are actually many different data types that are pointers, depending on the data type of the value stored in the memory location to which the pointer points. For example, an **`int`** pointer points to an **`int`** value in memory, a **`double`** pointer points to a **`double`** value stored in memory.

Declaring Pointers

Unlike other data types in C++, the pointer types aren't given a specific name. This means that they are declared in a slightly different way. The syntax to declare a pointer is

*dataType *identifier;*

To declare an **`int`** pointer you would have

```
int *k;
```

And to declare a **`double`** pointer, you would have

```
double *y;
```

In the above two examples, the asterisk (`*`) indicates that the identifier is a pointer to the specified type.

All three of the following declarations are equivalent:

```
int *k;

int* k;

int  *  k;
```

In the following statement, only **`i`** is a pointer. Variable **`j`** is an **`int`**, not an **`int`** pointer.

```
int* i, j;
```

To declare both **`i`** and **`j`** as **`int`** pointers, you would have to do this

```
int *i, *j;
```

Assigning Values to Pointers

Since a pointer holds a memory location, we use the address of operator **&** to assign a value to a pointer, like this

```
int x = 7; int *p = &x;
```

Here, **x** is an **int** variable and **p** is a pointer that holds the memory location where the value of **x** is stored. We say the **p** points to **x**. To access the value of **x** using the pointer **p**, we dereference the pointer like this

```
cout << *p;
```

Which causes the value of **x** to be displayed. The program in Listing 8.1 and its output in Figure 8.1 demonstrate how a pointer works.

Programming Example: Pointers1.cpp