University of Anbar
College of Computer Science and Information Technology
Department of Computer Networks

Pointers

Lecture 7

Dr. Ali Al-Kubaisi

Asst.Lect. Abdulrahman Abbas Mukhlif

Outlines

- C++ References
  - Creating References
  - Memory Address
- C++ Pointers
  - Creating Pointers
  - Get Memory Address and Value
  - Modify the Pointer Value

C++ References

Creating References

- A reference variable is a "**reference**" to an **existing variable**, and it is created with the **&** operator:

```
string food = "Pizza";  // food variable
string &meal = food;    // reference to food
```

- Now, we can use either the variable name food or the reference name meal to refer to the food variable:

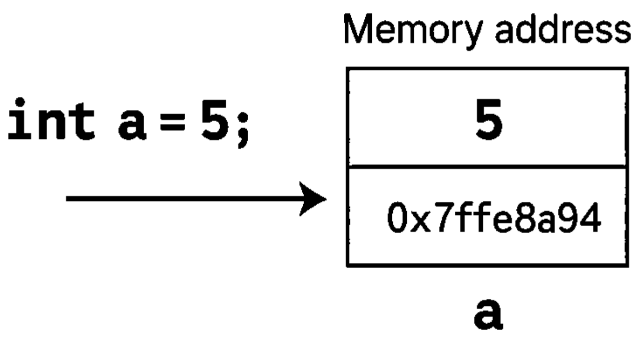
```
cout << food << "\n";  // Outputs Pizza
cout << meal << "\n";  // Outputs Pizza
```

Memory Address

- The **&** operator was used to create a reference variable. But it can also be used to get the **memory address of a variable**; which is the *location of where the variable is stored on the computer*.
- When a variable is created in C++, a memory address is assigned to the variable. And when we assign a value to the variable, it is stored in this memory address.
- To access it, use the **&** operator, and the result will represent where the variable is stored:

```
int a= 5;

cout << &a; // Outputs 0x7ffe8a94
```



The memory address is in hexadecimal form (0x..). Note that you may not get the same result in your program.

C++ Pointers

Creating Pointers

- A **pointer**, is a variable that **stores the memory address as its value**.
- A pointer variable points to a data type (like int or string) of the same type, and is created with the * operator. The address of the variable you're working with is assigned to the pointer:

```
string food = "Pizza";  // A food variable of type string
string* ptr = &food;// A pointer variable, with the name ptr, that stores the address of food

// Output the value of food (Pizza)
cout << food << "\n";

// Output the memory address of food (0x6dfed4)
cout << &food << "\n";

// Output the memory address of food with the pointer (0x6dfed4)
cout << ptr << "\n";
```

Creating Pointers (Cont.)

- There are three ways to declare pointer variables, but the first way is preferred:

```
string* mystring; // Preferred
string *mystring;
string * mystring;
```

Get Memory Address and Value

- We used the pointer variable to get the memory address of a variable (used together with the & **reference** operator).
- However, you can also use the pointer to get the value of the variable, by using the * operator (the **dereference** operator):

```
string food = "Pizza";  // Variable declaration
string* ptr = &food;    // Pointer declaration

// Reference: Output the memory address of food with the pointer (0x6dfed4)
cout << ptr << "\n";

// Dereference: Output the value of food with the pointer (Pizza)
cout << *ptr << "\n";
```

- Note that the * sign can be confusing here, as it does two different things in our code:
  - When used in declaration (string* ptr), it creates a **pointer variable**.
  - When not used in declaration, it act as a **dereference operator**.

Modify the Pointer Value

- We can change the pointer's value. But note that this will also change the value of the original variable:

```
string food = "Pizza";
string* ptr = &food;

// Output the value of food (Pizza)
cout << food << "\n";
// Output the memory address of food (0x6dfed4)
cout << &food << "\n";
// Access the memory address of food and output its value (Pizza)
cout << *ptr << "\n";
// Change the value of the pointer
*ptr = "Hamburger";
// Output the new value of the pointer (Hamburger)
cout << *ptr << "\n";
// Output the new value of the food variable (Hamburger)
cout << food << "\n";
```