2.17: Unary Operations

Overview

A **unary operation** is an operation with only one operand. As unary operations have only one operand, they are evaluated before other operations containing them.\(^1\) Common unary operators include Positive (+) and Negative (−).

Discussion

Unary positive also known as plus and unary negative also known as minus are unique operators. The plus and minus when used with a constant value represent the concept that the values are either positive or negative. Let’s consider:

\[ +5 + -2 \]

We have three operators in this order: unary positive, addition, and unary negative. The answer to this expression is a positive 3. As you can see, one must differentiate between when the plus sign means unary positive and when it means addition. Unary negative and subtraction have the same problem. Let’s consider:

\[ -2 - +5 \]

The expression evaluates to negative 7. Let’s consider:

\[ 7 - -2 \]

First constants that do not have a unary minus in front of them are assumed (the default) to be positive. When you
subtract a negative number it is like adding, thus the expression evaluates to positive 9.

---

**Negation – Unary Negative**

The concept of negation is to take a value and change its sign, that is: flip it. If it is positive make it negative and if it is negative make it positive. Mathematically, it is the following C++ code example, given that money is an integer variable with a value of 6:

\[-money\]

\[money \times -1\]

The above two expressions evaluate to the same value. In the first line, the value in the variable money is fetched and then it's negated to a negative 6. In the second line, the value in the variable money is fetched and then it's multiplied by negative 1 making the answer a negative 6.

---

**Unary Positive – Worthless**

Simply to satisfy symmetry, the unary positive was added to the C++ programming language as on operator. However, it is a totally worthless or useless operator and is rarely used. However, don’t be confused the following expression is completely valid:

\[6 + +5\]

The second + sign is interpreted as unary positive. The first + sign is interpreted as addition.

\[money\]

\[+money\]

\[money \times +1\]

For all three lines, if the value stored in money is 6 the value of the expression is 6. Even if the value in money was negative 77 the value of the expression would be negative 77. The operator does nothing because multiplying anything by 1 does not change its value.

---

**Possible Confusion**

Do not confuse the unary negative operator with decrement. Decrement changes the value in the variable and thus is an Lvalue concept. Unary negative does not change the value of the variable but uses it in an Rvalue context. It fetches the value and then negates that value. The original value in the variable does not change.

Because there is no changing of the value associated with the identifier name, the identifier name could represent a variable or named constant.
Exercises
Evaluate the following items involving unary positive and unary negative:

1. \(+10 - -2\)
2. \(-18 + 24\)
3. \(4 - +3\)
4. \(+8 + - +5\)
5. \(+8 + / +5\)

Key Terms

minus  
Aka unary negative.

plus  
Aka unary positive.

unary negative  
An operator that causes negation.

unary positive  
A worthless operator almost never used.

References

- cnx.org: Programming Fundamentals – A Modular Structured Approach using C++

Wikipedia: Unary operation