COEN-244
Tutorial #7

March 9th, 2023

Operators

In **C++,** the definitions of the operators are defined for the different datatypes:

- E.G., the **'+'** operator does **addition** on integers and **concatenation** for strings
- Operators reduce the ambiguity in the code and they offer useful operations

Operators in C++ can be classified into 6 types:

- Arithmetic Operators
- Assignment Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Other Operators

Types of Operators

Arithmetic: +, -, *, /, %, ++, --

Assignment: =, +=, -=, *=, /=, %=

Relational: ==, !=, >, <, >=, <=

Logical: &&, ||, !

Bitwise: &, |, ^, ~, <<, >>

Other: sizeof(), ?:, &, ->, <<, >>

Operator Overloading: Why?

OPERATOR OVERLOADING: Defining/Re-defining the operations an operator should do for user-defined data types or objects.

- They are like specialized functions that are in the form of operators.

SCENARIO:

Say there is a class `Shape`, and we want to output a shape object `(rectangle)` in the following form:

- `cout << rectangle << endl;` (What happens?)
- Hence, we have to give it a definition in the class `Shape`;

WHY?

Generally, it is encouraged to be explicit in the our code, **i.e.**, explicitly defining what each component does instead of relying on the compiler.

Overloadable Operators

+	-	*	/	%	^
&	|	~	!	,	=
<	>	<=	>=	++	--
<<	>>	==	!=	&&	||
+=	-=	/=	%=	^=	&=
|=	*=	<<=	>>=	[ ]	()
->	->*	new	new [ ]	delete	delete [ ]

Operator Overloading Exercise

EXERCISE 1:

i. Overload the division '/' operator for the `Fraction` class.

ii. Overload the `ostream` `<<` operator for printing a fraction object in a suitable form.

EXCERISE 2:

i. Overload the assignment '=' operator for matrices

ii. Overload the `ostream` `<<` operator for printing a matrix object in a suitable form.

Files: https://github.com/TheBarzani/COEN244_W2023/tree/main/tut7

THANK YOU