Operators

An operator is a symbol that tells the compiler to perform specific mathematical operation and produce final result.

Operand	Operator	Operand
1	+	2

The following are the types of operators in Rust:

- Arithmetic Operators
- Comparison Operators
- Logical Operators
- Bitwise Operators (not included in this course)
- Compound Assignment Operators

Arithmetic Operators

Arithmetic operators are used for performing mathematical operations like addition, subtraction, multiplication, and division.

Operator	Explanation	Example
+	Returns the addition of two operands	A+B = 10
-	Returns the difference of the values (subtract right operand from left)	A-B = 23
*	Returns the product of the values	A*B = 100
/	Divide left operand by right one and returns the quotient	A/B = 3
%	Divide left operand by right one and returns the remainder	A%B = 9

Comparison Operators

Comparison operators are the operators that compare values and return true or false depending upon the conditions.

Operator	Explanation	Example
>	Greater than	(A+B) is true
<	Less than	(A<B) is false
==	Equal to	(A==B) is false
!=	Not equal to	(A!=B) is true
>=	Greater than and equal to	(A>=B) is true
<=	Less than and equal to	(A<=B) is false

Logical operator

Logical Operators are used to combine two or more conditions. Logical operators too, return a Boolean value.

Operator	Explanation
&&	The operator returns true only if all the expressions specified return true
||	The operator returns true if at least one of the expressions specified return true
!	The operator returns the inverse of the expression's result

Compound Assignment Operator

Compound-assignment operators perform the operation specified by the additional operator, then assign the result to the left operand.

Operator	Explanation
+=	Arithmetic addition and assignment
-=	Arithmetic subtraction and assignment
*=	Arithmetic multiplication and assignment
/=	Arithmetic division and assignment
%=	Arithmetic remainder and assignment