# C++

1. #include<iostream>
2. using *namespace* std;
3. *int* main()
4. {
5. *int* n = 1;
6. *char* c = 'a';
7. *float* f = 3.8;
8. string str1 = "Hello";
9. string str2 = "World";
10. cout<<"int: "<< n <<" + float: "<<f <<"= "<<f+n<<endl;
11. cout<<"char: "<< c <<" + int: "<<n <<"= "<<c+n<<endl;
12. cout<<"string: "<< str1 <<" + string: "<<str2 <<"= "<<str1+str2<<endl;
13. return 0;
14. }

Output:

A black background with white text

Description automatically generated

**Key Points**:

1. Integer and float addition uses Implicit Coercion. Integer is converted to float before addition.
2. Character and integer addition is allowed in a sense that the character is converted to its integer ascii value and added to the integer.
3. String + string implicit concatenation is allowed.

# 2. Python

n = 1

c = 'a'

f = 3.8

str1 = "Hello"

str2 = "World"

print("int: ", n, "+ float: ", f, "= ", n + f)

print("char: ", c, "+ int: ", n, "= ", ord(c) + n)

print("string: ", str1, "+ string: ", str2, "= ", str1 + str2)

A black background with white text

Description automatically generated

**Key Points:**

1. Python automatically handles type coercion when adding integers and floats.
2. ord(c) converts the character 'a' to its ASCII value (97), and it is then added to the integer 1.
3. String concatenation is done using the + operator.

# 3. Java

package assignment\_2;

public class Test  {

    public static *void* main(String[] *args*) {

*int* n = 1;

*char* c = 'a';

*float* f = 3.8f;

        String str1 = "Hello";

        String str2 = "World";

        System.out.println("int: " + n + " + float: " + f + " = " + (n + f));

        System.out.println("char: " + c + " + int: " + n + " = " + (c + n));

        System.out.println("string: " + str1 + " + string: " + str2 + " = " + str1 + str2);

    }

}

A black background with pink and white numbers

Description automatically generated

**Key Points:**

1. Java performs implicit type conversion when adding an int to a float.
2. When adding a character to an integer, Java converts the character to its ASCII value.
3. Strings are concatenated using the + operator.

# 4. Golang

package main

import (

    "fmt"

)

func main() {

    n := 1

    c := 'a'

    f := 3.8

    str1 := "Hello"

    str2 := "World"

    fmt.Printf("int: %d + float: %f = %f\n", n, f, *float64*(n)+f)

    fmt.Printf("char: %c + int: %d = %d\n", c, n, *int*(c)+n)

    fmt.Printf("string: %s + string: %s = %s\n", str1, str2, str1+str2)

}

A black screen with white and pink numbers

Description automatically generated

**Key Points:**

1. In Go, int needs to be explicitly converted to float64 for the addition to work.
2. The character c is converted to its Unicode code point before adding it to int.
3. String concatenation uses the + operator.

# 5. Ruby

n = 1

c = 'a'

f = 3.8

str1 = "Hello"

str2 = "World"

puts "int: #{n} + float: #{f} = #{n + f}"

puts "char: #{c} + int: #{n} = #{c.ord + n}"

puts "string: #{str1} + string: #{str2} = #{str1 + str2}"

A black background with white and pink numbers

Description automatically generated

**Key Points:**

1. **Integer and Float Addition**: Ruby automatically handles type conversion when adding an integer to a float.
2. **Character and Integer Addition**: In Ruby, c.ord converts the character to its ASCII value (similar to C++), which can then be added to an integer.
3. **String Concatenation**: The + operator is used for string concatenation in Ruby.