Class Exercise:

Interop v1

In this exercise, we will explore the interoperability between Python and C using two popular libraries: Cython and ctypes. The goal is to demonstrate how we can call C functions from Python and vice versa.

The exercise will involve creating a Python class that uses Cython to call a C function and retrieve a result. The C function will perform a simple mathematical operation, such as adding two numbers, and return the result. We will then use ctypes to call the Python class from a C program and print the obtained result.

Exercise Solution:

First, let's create a Python class that uses Cython to interact with a C function:

# interop.pyx

cdef extern from "c\_functions.h":

    int add\_numbers(int a, int b)

cdef class Interop:

    cdef int a

    cdef int b

    def \_\_init\_\_(self, a, b):

        self.a = a

        self.b = b

    def add(self):

        return add\_numbers(self.a, self.b)

Next, we need to create a C header file that declares the add\_numbers function:

// c\_functions.h

int add\_numbers(int a, int b);

Now, let's create a C program that uses ctypes to call the Python class and print the result:

// main.c

#include <stdio.h>

#include <dlfcn.h>

typedef int (\*AddFunc)(int, int);

int main() {

    void\* handle = dlopen("./interop.so", RTLD\_LAZY);

    if (!handle) {

        fprintf(stderr, "Error: %s\n", dlerror());

        return 1;

    }

    AddFunc add\_numbers = (AddFunc)dlsym(handle, "Interop\_add");

    if (!add\_numbers) {

        fprintf(stderr, "Error: %s\n", dlerror());

        dlclose(handle);

        return 1;

    }

    int a = 5;

    int b = 7;

    int result = add\_numbers(a, b);

    printf("Result: %d\n", result);

    dlclose(handle);

    return 0;

}

To compile the C program, we need to create a Makefile:

# Makefile

all: main.c

    gcc -o main main.c -ldl

Once you have created these files, follow these steps to run the exercise:

1. Save the interop.pyx file.
2. Compile the Cython file to generate the shared object file:

cythonize -i interop.pyx

This will generate the interop.so file.

1. Create the c\_functions.h file and save it in the same directory.
2. Create the main.c file and save it in the same directory.
3. Create the Makefile and save it in the same directory.
4. Compile the C program using the Makefile:

make all

1. Run the C program:

./main