Hello Again!

Due Tuesday, April 1 at 8 a.m.

CSE 1325 - Spring 2025 - Homework #8

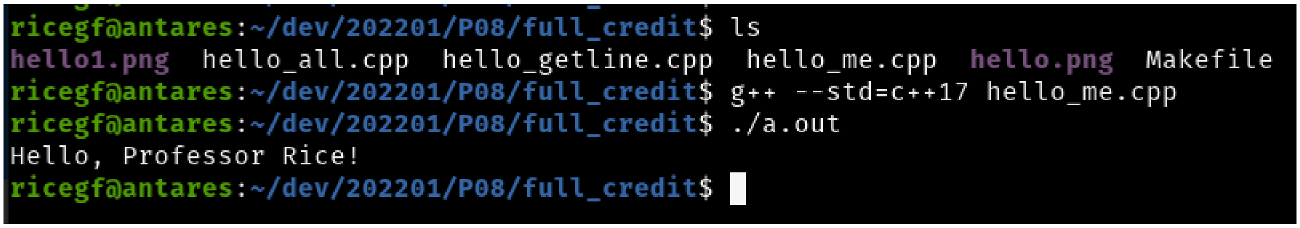
Assignment Overview

Way back in homework P01, you wrote the classic "Hello, World!" program in Java. Now that we're switching our focus to C++, let's retread that hallowed ground!

Full Credit

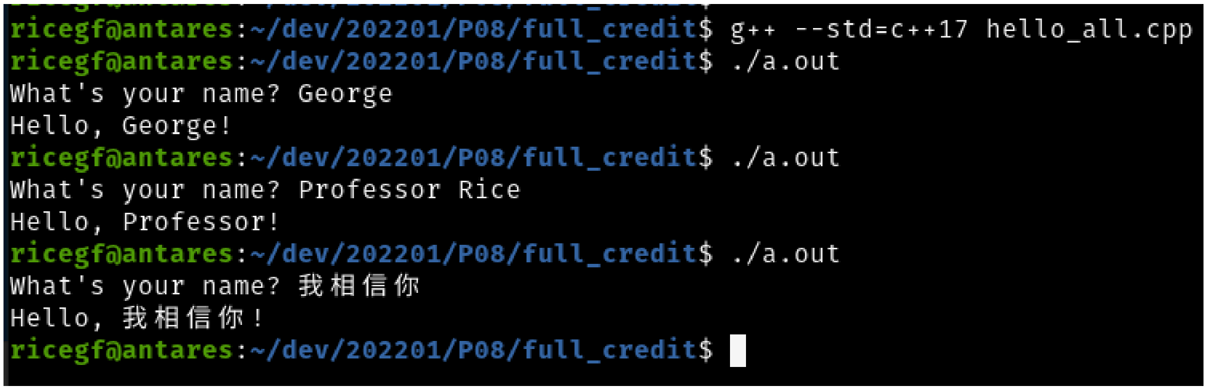
For this full credit assignment, we will repeat both the P01 full credit and bonus assignments - in C++!

If you haven't already, set up your C++ development environment using guidance from Lecture 00 and the current slides. Then in **cse1325/P08/full_credit**, create and run the C++ “Hello, World!” program **in file hello_me.cpp** using `std::cout`, **replacing “World” with your name**. Take a screenshot of your program running, and **name it hello_me.png**.



```
ricegf@antares:~/dev/202201/P08/full_credit$ ls
hello1.png  hello_all.cpp  hello_getline.cpp  hello_me.cpp  hello.png  Makefile
ricegf@antares:~/dev/202201/P08/full_credit$ g++ --std=c++17 hello_me.cpp
ricegf@antares:~/dev/202201/P08/full_credit$ ./a.out
Hello, Professor Rice!
ricegf@antares:~/dev/202201/P08/full_credit$ █
```

Then create and run a C++ program **in file hello_all.cpp** that **asks the user for their name**, then prints “Hello, [name]”, where [name] is the name they enter. Test this at least 3 times - with a single-word name, with a mult-word name separated by spaces, and by a name with different (potentially weird) characters. Take a screenshot of your program running all of these tests, and **name it hello_all.png**. As long as your program prints *something* after "Hello, ", you get full credit regardless of the outcome of these 3 or more tests.



```
ricegf@antares:~/dev/202201/P08/full_credit$ g++ --std=c++17 hello_all.cpp
ricegf@antares:~/dev/202201/P08/full_credit$ ./a.out
What's your name? George
Hello, George!
ricegf@antares:~/dev/202201/P08/full_credit$ ./a.out
What's your name? Professor Rice
Hello, Professor!
ricegf@antares:~/dev/202201/P08/full_credit$ ./a.out
What's your name? 我相信你
Hello, 我相信你!
ricegf@antares:~/dev/202201/P08/full_credit$ █
```

Add hello_me.cpp, hello_me.png, hello_all.cpp, and hello_all.png to git. Commit and push your work to GitHub.

Bonus

In **cse1325/P08/bonus**, write a C++ main function in file sorted.cpp that converts the standard `char* argv[]` main parameter into an array of `std::string`.

The *sort* the array of `std::string` using `std::sort` (look it up in the documentation, and feel free to check online for examples). Hint: This is actually much easier than sorting `argv[]` itself!

Finally, use a for-each loop (yes, C++ has them, too!) to print out the sorted array of `std::string`.

That is,

```
$ ./sorted my fake plants died because I did not pretend to water them
```

would print

```
I
because
did
died
fake
my
not
plants
pretend
to
water
them
```

No screenshots are required. Add, commit, and push all files.

Hint: `std::string` has a constructor that will accept a `char*`.

Hint: In Java, variables *reference* an object such as a `String`, so assigning `a = b;` just creates another reference to the *same* object.

But in C++, stack variables *contain* an object such as `std::string`, and so `a = b;` makes a *copy* of the object. This is what you usually want in C++.

Hint: In C++, == compares the *value* of both primitives and objects alike - even `std::string` objects! C++ has no `.equals` method. (You would compare the *address* of objects exactly as in C, of course: `&m1 == &m2`.)