Assignment 6
cpe 453 Winter 2026

```
It is generally agreed that "Hello" is an appropriate greeting because
if you entered a room and said "Goodbye," it could confuse a lot of people.
                -- Dolph Sharp, "I'm O.K., You're Not So Hot"
```

— /usr/games/fortune

Due by 11:59:59pm, Friday, March 13th.
This assignment is to be done individually.

Program: `hello`

Write a program that greets the world by printing "`Hello, world!`" and a Makefile that compiles and tests it.
Your program must:

1. Greet the world by printing "`Hello, world!`" followed by a newline.

2. Exit with a zero exit status indicating success.

Your Makefile must:

1. properly build the program `hello` either when given no target or when given the target "`hello`"

2. remove all extraneous object files when given the target "`clean`."

3. build and execute the program `hello` when given the target "`test`."

You must:

1. spend a few minutes reflecting upon and appreciating everything that goes into the loading and running of a program, even one as simple as this. Think of how everything we have done this quarter applies.

Tricks and Tools

I can't think of anything to suggest here other than to actually read the whole assignment and follow the specifications.

Well, there is one thing: if you want to use a command in a Makefile that you don't want echoed, you can preceed it with a '@' and it will be executed without being echoed.

The Catch

It really has to be right. This means:

- Your program must follow conventions for proper style and documentation (yes, really).

- Your program must compile cleanly under `gcc -Wall -ansi -pedantic` on my machine. (It shouldn't be too hard to write a portable "Hello, world!" after all.)

- The Makefile needs to correctly build, test, and clean as specified above.

Coding Standards and Make

See the pages on coding standards and make on the cpe 453 class web page.

What to turn in

Submit in the CSL via `handin` to the `asgn6` directory of the `pn-cs453` account:

- Your well-documented source file, `hello.c`.

- A makefile (called `Makefile`) as specified above.

- A README file that contains:

  - Your name.
  - Any special instructions for running your program.
  - Any other thing you want me to know while I am grading it.

Sample runs

Below are some sample runs of `hello`.

```
% ls
Makefile  RCS/  hello.c
% make hello
gcc -Wall -ansi -pedantic -g   -c -o hello.o hello.c
gcc -g -o hello hello.o
% ./hello
Hello, world!
% rm hello
% ls
Makefile  RCS/  hello.c  hello.o
% make
gcc -g -o hello hello.o
% ./hello
Hello, world!
% make clean
rm -f hello.o *~ TAGS
% ls
Makefile  RCS/  hello   hello.c
% make test
gcc -Wall -ansi -pedantic -g   -c -o hello.o hello.c
gcc -g -o hello hello.o
Testing hello...
Hello, world!
done.
%
```