

# Homework 1 - Artificial Intelligence (sort of...)

Jee Whan Choi & Chris Misa

April 5, 2019

**DUE DATE: 11:59 PM 4/12/2019**

The objective of today's homework is to implement an auto-response bot. Please read this document completely before starting your assignment.

Follow the instructions below to get started:

- Once the tarball has been extracted, you should see two files. This pdf, and hw01\_robot\_skeleton.c.
- Compile the code using “gcc hw01\_robot\_skeleton.c” which will generate the “a.out” executable.
- Try typing “./a.out Hello World” and see what is printed.
- It should print a number of messages - the tokens “Hello” and “World” and some string comparisons and more.
- Study the code to see what is happening. This is **important** since you will use some or all of these functions to implement the homework.

In the source, there is a variable named “INPUT\_STR.” This variable contains a list of words (e.g., “Good,” “Leaving,” etc.) which your robot **should recognize** when you enter it as input to your program (e.g., we entered “Hello World” in the above instruction). If the robot sees any of these words, it should respond (i.e., print to the screen) with an appropriate message. These messages can be found in the variable “RESPONSE\_STR.” If the robot recognizes that the user entered a word in INPUT\_STR[n], it should print the message in RESPONSE\_STR[n], where n is a number from 0 to NUM\_RESPONSE - 1, inclusive.

Things to note:

- If the user entered message matches multiple words in `INPUT_STR`, then the robot should respond with multiple corresponding messages from `RESPONSE_STR`.
- If the user entered messages matches **no** words from `INPUT_STR`, then it should print the last message in `RESPONSE_STR`, “I don’t understand what you are saying.” Note that `RESPONSE_STR` has one more message than there are words in `INPUT_STR` to respond when it sees no recognizable words.
- The user entered word can be in upper case, lower case, or any combination thereof. The robot should be agnostic to the case (i.e., it is **not** case sensitive).
- Enter your code where it says “Enter your code here.” If you do not, your output may appear outside the “— Answer —” and “——” messages, and this may cause the grader to give 0 for the assignment.

Below are some example inputs and how the robot **should** respond when properly implemented.

- `./a.out A Whole New World`  
I don’t understand what you are saying.
- `./a.out GooD Day to YOU`  
Good day to you as well.
- `./a.out Good Leaving Age Weather Sports`  
Good day to you as well.  
Good bye.  
I am a robot and I am 4 hours old.  
It’s raining.  
Go Ducks!

When you have completed your assignment, verify it on ix-dev and then rename the file to `hw01_robot_answer.c` before adding, committing, and then pushing the file to your Bitbucket repository. Extra credit will be given if you can find ways to minimize the amount of work (i.e., comparisons) that must be done to find your answer.

Have fun with your assignment and don’t hesitate post questions on Piazza if something is ambiguous.