

Home » COMP1730/6730: Assessment » Homework 1 (S1 2019)

Homework 1 (S1 2019)

Homework 1 (S1 2019)

This is the first homework assignment. Your goal in this assignment is to read and understand a simple program, and modify it so that it works. Along the way, it is also important that you get familiar with the python programming environment, and take your first step towards learning how to use it effectively.

Practical information

The assignment is due on **Sunday the 10th of March, at 11:55pm**. To submit your solution, you will upload a single python file via wattle. Here is the assignment submission link.

In addition to submitting your solution, you must attend the following lab (in week 3). In the lab, your tutor will ask you some questions about your solution, and give you feedback if there is anything you need to improve. This discussion with the tutor is also part of the assessment. If you do not show up for the discussion with the tutor, you will not receive any marks for this assignment. If you do not submit a solution, you may still get some partial marks following the discussion with the tutor.

The homework is individual. You must write your own solution, and you are expected to be able to explain every aspect of it.

If you have followed the lectures and worked through the exercises in lab 1, the assignment should not take more than one to two hours to complete. There will also be a survey on wattle, asking you some questions about the assignment (for example, how long you spent on it). The survey is not part of the assessment.

The problem

This assignment problem uses the robot simulator that was also used in lab 1. First, you must download these two files:

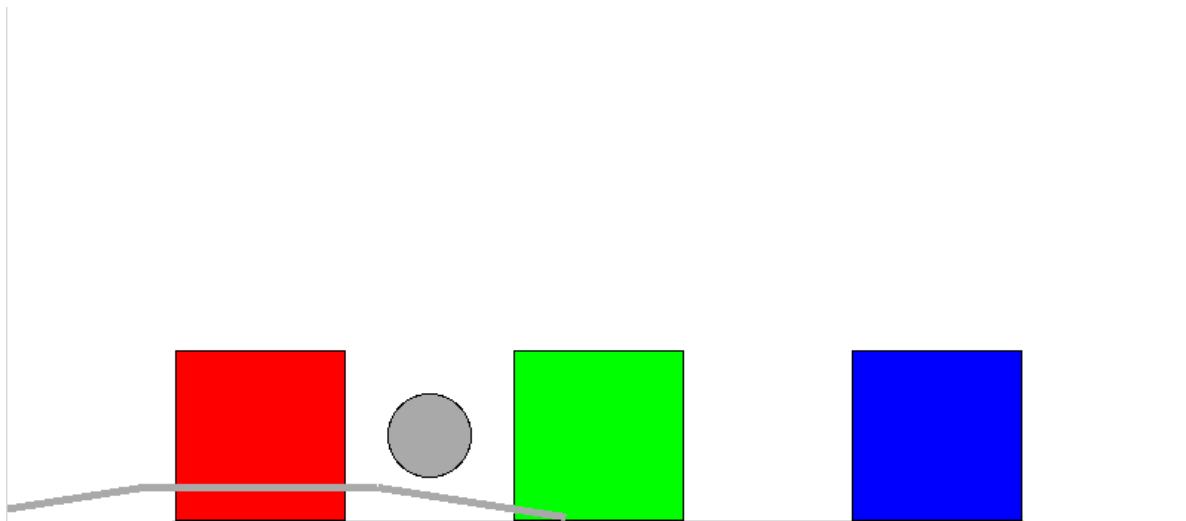
- robot.py
- unstack.py

Save both files to the same directory.

The problem is how to unstack a tower of blocks. That is, to go from this:



to this:



The colours of the blocks are not important: After unstacking, the blocks should be ordered left-to-right in the same order that they were top-to-bottom in the tower, no matter what their colours are. The final position of the robot also does not matter.

`unstack.py` contains several functions that implement a solution to this problem. Because the robot (at least in simulation) can carry more than one block, the idea of the solution is that we will first unstack the top two blocks from the third, put down this stack of two, and then solve the smaller problem of unstacking the top block from the stack of two using the same manoeuvre. However, the implementation of this solution has some errors. Your task is to correct the errors.

`unstack.py` also contains two test functions, called `test_unstack_two()` and `test_unstack_three()`. The first tests the simpler case of unstacking just two blocks, while the second tests unstacking of a tower of three.

Because the file contains only function definitions, you will not see anything happen when you run it. To run the test functions, you must first run the file and then call the function, for example

In `[1]: test_unstack_two()`

(Remember that if you run the python interpreter from a terminal, you have to use the `-i` flag to leave the shell in interactive mode after running the file.)

The first error you will encounter is a `NameError`. As mentioned in the lecture, this means there is a mismatch between the name of a function's definition and use. Use the information from the python error message to track down where the error is.

Once you have resolved this error, the `test_unstack_two()` function should work. In the solution for unstacking three blocks there is a logical error that may require more thought to fix. However, it does not require a big change to the functions. You should not write an entirely new program. Focus on understanding how the given program is intended to work, and what changes are needed to make it work. (It may be as simple as changing one line.)

Marking

What to submit

You should edit the file `unstack.py` that you downloaded, then upload only this file with your corrections using the assignment submission link on wattle. Do not edit (and do not try to upload) `robot.py`.

The file that you submit must meet the following requirements:

- It must be syntactically correct python code.
- Like the file you downloaded, it must contain **only function definitions, the `import` statement, and comments** (module and function docstrings are accepted). If you are not sure what is accepted, ask your tutor during the lab.
- It must contain the two functions `test_unstack_two()` and `test_unstack_three()`, and these functions **must not be modified** (that is, they should be exactly the same as in the file you downloaded).

As mentioned above, you must also attend the following lab (in week 3) and answer your tutor's questions about your solution. This discussion is part of the assessment. You should be prepared to answer or demonstrate to the following questions:

- Can you download the file that you submitted from wattle?
- Can you run that file in the python interpreter (using an IDE of your choice) on the CSIT lab computer?
- If the file has syntax errors, can you use the error messages from the interpreter or IDE to identify where the syntax errors are?
- Does your submitted file meet the requirements stated above? Does it contain anything that is not a function definition, an `import` statement, or a comment? If so, can you point it out?
- What was the cause of the `NameError` in the original file, and how did you fix it?
- How did you fix the logical error in the `unstack_tower_of_three` function? Was this the smallest change necessary to fix the error? Could it have been done with a smaller change?
- Can the solution to the unstacking problem be made more efficient, that is, with fewer moves of the robot or gripper?
- The original file had comments describing the each function, and the assumptions about the state of the robot when the function is called. Are these comments still accurate in your version? Have you modified the comments so that they correctly describe the functions that you modified?

In marking this assignment we will consider the following:

- Does your submitted file satisfy the requirements specified above? (Submission that do not meet the syntactic requirements **can not receive more than half the marks.**)
- Do the two test cases (`test_unstack_two` and `test_unstack_three`) work with your modified solution?
- Your ability to use the tools (e.g., the IDE or python interpreter), your understanding of python's error messages, and your understanding of the solution, as demonstrated in your discussion with the tutor.

The assignment is worth 2% of your final mark.