# T4: Unit Testing

* Paired assignment T4 should be completed with a partner.
* To begin, go to “File” and Select “Make a Copy...”
* Move your copy of the document to the folder for your section.
* Rename the file to **T4: Unit Testing - usernames** (replace usernames with your usernames). To do this, click the label in the top left corner of your browser.

## Learning Objectives

* Understanding the value of thorough testing.
* Applying unit testing to code to root out bugs.
* Explore the modulus operator.
* Explore Conditionals in more detail.

First, discuss with your team and assign yourselves roles. Try to pick the role you’ve had the least experience in.

|  |  |
| --- | --- |
| **Driver[[1]](#footnote-0):** |  |
| **Navigator[[2]](#footnote-1):** |  |
| **Quality Control[[3]](#footnote-2) (if the class is odd numbered):** |  |

### The modulus (%) operator

To learn about the modulus operator, consider this program: <https://goo.gl/RdzO4N>

Using the "Forward>" button again, step through each line of code, and briefly explain what each step does.

|  |  |
| --- | --- |
| Explain what the program does, in plain English. | 1. |
| In which steps do x and y exit? Where do these values first come from? | 2. |
| Explain why  remainder = x % y  followed by  if remainder == 0:  gives information about if y goes into x evenly. | 3. |
| In which steps does the variable remainder exit? Where does this value first come from? | 4. |
| Explain how the following code works:  if is\_divisible(n\_val, d\_val)  in the way we would hope. | 5. |

## Unit Testing

It is considered a best practice in software development to include automatic "unit testing" of source code. **Unit testing** simply means systematically testing of each logical unit or logical block of code. This provides a way of verifying that these individual blocks of code, such as functions, are working properly. This is a good method because each **test suite** of unit tests can be preserved, making it possible to change the implementation of a function at a later time and still easily and quickly test that it still does what it was intended to do by making sure it still satisfies its unit tests.

Many organizations see both code and program documentation as valuable assets. Recently, suites of unit tests have been preserved to document the suite of tests used to test each function's correctness.This suite can be stored right in the code base long after the testing phase is complete. In fact, extra code in your program which is only there because it makes debugging or testing easier is called **scaffolding**.

Some errors are subtle. You should not feel bad when your code has errors. All programmers make errors. However, by becoming a strategic sleuth and documenting your sleuthery using unit tests, you can generally eliminate the errors.

Let’s look at an example program which use a test suite with unit tests:

* [t4\_is\_even.py](https://drive.google.com/open?id=0B0J8Yj0B6KRSd2gwdHBiSXk2NjA)

Please try to run and understand the flow of execution in the above programs as the test suites run. Note also that the unit tests are organized into blocks and those blocks are commented. For example, in testing whether an integer is even, we want to test large and small positive numbers, zero, and negative numbers which are large and small in magnitude.

## Making Change

The point of this part of the assignment is NOT to fix the code. **DO NOT MODIFY ANY OF THE EXISTING CODE!** Your job, as far as the code goes, is ONLY to add an appropriate test-suite.

This is because the primary objective is to learn to use unit tests effectively. Some errors are really subtle. I am giving you one of these subtle errors. Sometimes in testing you can feel like you are going around and around in circles or banging your head against the wall. Unit tests are designed to help you become a detective and to remember what you have already ruled out.

|  |  |
| --- | --- |
| Download the following buggy python code sample: [t4\_making\_change.py](https://drive.google.com/open?id=0B0J8Yj0B6KRSakFwNVpjUWJ2VEU)  In your own words, describe what does the i\_steal\_pennies() function do? | 6. |
| Does the i\_steal\_pennies() function appear to be coded correctly? | 7. |
| Your task is to create more unit tests (line 53) which would help you to sleuth out categories of numbers for which the unit tests pass and categories in which the unit tests fail. Organize your unit tests into categories of inputs and add comments to each section regarding your thinking about each category.  **Create as many unit tests as you feel are necessary to correctly identify the categories of numbers in which the given i\_steal\_pennies() function fails to make correct change.** This may be a large number of unit tests. That is good!  In the space to the right, paste in all of the unit test values that fail. I’ve given you the first example: | 8. i\_steal\_pennies(0.99)==[3, 2, 0, 4] |
| Explain why this line of code works:  if did\_pass: | 9. |
| If you are extra-specially mathematically inclined, you might even be able to describe the category of numbers for which the program steals pennies. If you can, describe these numbers in the comments. This is not required, but I will be impressed if you can do it! | 10. |

## Submission Instructions

1. (Submitter) Save your initials code as **T4\_unit\_testing\_usernames.py**. Replace *usernames* with your Berea usernames. For example, the TA Bianca Marrero and my document would be named **T4\_unit\_testing\_heggens\_marrerob.py.   
   NOTE:** Incorrect filenames will automatically reduce your grade by 1 point. Fortunately, the format is always the same no matter what the assignment.
2. (Submitter) Upload the Python file to Moodle by the due date listed on the course website: <https://trello.com/b/w7bIrLoV/>.
3. (All Other Partners) Open up Wordpad. Create a new text document (.txt) and include all members names in it.
4. (All Other Partners) Save the document as **T4\_unit\_testing\_usernames*.txt***. Replace *usernames* with your Berea usernames. For example, the TA Bianca Marrero and my document would be named **T4\_unit\_testing\_heggens\_marrerob.txt.**
5. (All Other Partners) Upload the document to Moodle by the due date listed on the course website: <https://trello.com/b/w7bIrLoV/>.

1. The driver will be doing the majority of the typing in PyCharm. Your job is to solve the problem given to you by the Navigator. [↑](#footnote-ref-0)
2. The navigator will be giving directions to the driver, and helping the driver catch syntax and logic errors as he or she creates the code. The navigator should keep track of time and make sure progress is being made. [↑](#footnote-ref-1)
3. The quality control specialist will ensure rules are being followed, both in the code (suggesting places to add comments, watching for misspellings, etc.) and in this document (making sure the questions are being answered at the right times, checking for typos, etc.) In a group of two, everyone is responsible for quality control. [↑](#footnote-ref-2)