CSCI 1100 — Computer Science 1
Fall 2017
General Comments: Read This Before Starting the Homework

Overview

Welcome to CS-1 homeworks. This short document should be used as a reference for all your homework submissions.

Submission of homeworks: test first, but you can submit many times

As you learned in Lab 1, all homework will be submitted electronically through *Submitty*, the Department of Computer Science homework submission and auto-grading server. This link is also available on the course website. The homework submission server for a particular assignment will typically be available no later than the Monday of the week the homework is due.

Make a habit of testing your program by running it using the Wing IDE. First, make sure that it can run. Then, go through the logic to make sure that it is correct. Learning to test your code is a big part of learning to program. Use Submitty, which can get slow near the deadline, for submitting and final changes rather than for testing. You will not be penalized for submitting a program multiple times, but you are generally graded on the last solution you submit. Note that the time you submit the last version determines whether a homework is late or not.

How will you be graded?

- Program correctness will be the most important determinant of your grade. In some cases, especially later in the semester, we will test your code with many different inputs, not just the ones we provide as sample test cases in the homework description.
- Small differences in output spacing will cause small losses of points. While we care mostly about correct logic, making sure the output matches the sample output exactly teaches you to use the `print` function and strings effectively.
- In addition to looking at output, we will also read your code. Make sure your program is well-written and has clear, correct program logic. Test it yourself with additional test cases, not only the ones we give you. You may lose points even if you match all test cases but your logic is unclear or incorrect.
- Most of your programs, particularly later in the semester, will ask you to implement somewhat complicated algorithms to simulate physics, to process cloud data, or to perform some other task. In these cases, should you get stuck or run out of time, you may be tempted to forget about the calculations and algorithms and just replace them with hardcoded `print` statements that mimic the expected output on `Submitty`. **Do not do it.** This is not a valid submission, the TAs are instructed to look for it, and you will get a **0**.
- We will look at programming style including the organization of your code, the written comments describing the purpose of your code, the names of variables, the use of functions, etc. This will be emphasized increasingly as we progress through the semester. It is important to develop good habits even when writing relatively short programs. Do not worry about this

for the first few homeworks. We will discuss it in class and provide you with guidelines before we start grading you on it, but be aware that it is be coming.

- Remember that generally the last version you submitted is the one that will be graded. You can use Submitty to make a past submission the active one if you want us to grade that one instead, but you need to specify this explicitly on the server. This must be done by the submission deadline. It is your responsibility to manage it.
- We often receive the question, **can I use a programming construct that we have not learned yet?** While you may not lose points for doing so — unless we give you an explicit warning — we strongly urge you not to. We design homeworks to specifically target the concepts we are learning right now. Hence, while there may be more advanced methods to solve a problem, if you feel you need to use them you are not properly learning and applying the concepts we are teaching.

Late homework policy

When a homework has multiple parts, the submission time is the time that the last part is submitted. For example, if you submit parts 1 and 2 three days before the due date and part 3 one hour after the submission deadline, you will have used up one late day. Reread the late homework policy in the syllabus. You have a total of three full or part days you can use for late homeworks in the whole semester and up to two days can be used for a single homework, assuming you have them available. It is highly recommended that you save these for later homeworks that will be harder to complete and longer.

Wing IDE vs. Homework Submission server

On *rare* occasions when the homework submission server runs your code it will produce different output than the Wing IDE does, even though both are using Python 3.5.2. In such cases pay careful attention to the server's output and try to figure out what happened, and remember that the input on `Submitty` may be different from the input you are given as part of the assignment. Think about what could be causing problems. Usually you've done something wrong in the way you wrote, submitted or executed your program. If you can't fix the problem, check the Piazza site for a relevant discussion. Only after checking should you post a Piazza question (and remember do not post your code!). This is also a good topic for help during office hours.

Input format for homeworks in this class

Your program must read the same number of inputs as are required according to the given problem. For example, if we tell you to read a name first and an email address next, that means your program must have two `input` statements. If it does not, you will get an error like:

```
EOFError:  EOF when reading a line
```

This means either you are trying to read too many or too few inputs. Read the problem specification carefully.

In all homeworks, we will use a convention specific to CS-1. If we ask you to read an input, then after you read it you must immediately print that input. For example, the following is the correct method to input the name string and the email address string:

```
name = input('Enter a name ==> ')
print(name)
email = input('Enter an email ==> ')
print(email)
```

The above program will produce a different output in Wing IDE and on Submitty, as shown below:

```
Enter a name ==> Rensselaer
Rensselaer
Enter an email ==> rpiinfo@rpi.edu
rpiinfo@rpi.edu
```

Wing IDE output
(what you see on your computer)

```
Enter a name ==> Rensselaer
Enter an email ==> rpiinfo@rpi.edu
```

Homework submission output
(what you see on submission server)

If you forgot to add the print function calls, you would actually see something like this in the submission server which will be considered incorrect output:

```
Please enter a name==> Please enter your email==>
```

The difference — and this is not really important for the first few homeworks — is that for each part of the homework, we place all the input into a file and do what's called "running from the command-line." In the above example, we are using an input file (let's assume it is called `input.txt`) that contains the two strings `Rensselaer` and `rpiinfo@rpi.edu` on two lines.

When a program is run from a command shell, we use a command-line of the form:

```
python part1.py < input.txt
```

Unfortunately, the free version of the WingIDE that we have been using in class does not allow us to specify this input. But, you can do the command-line form on Mac/Linux with a Terminal window, and on Windows using Linux-derived Cygwin tools. If you want to learn how, you can ask us during office hours. Anyway, here is the bottom line:

Anytime you read input, print it immediately afterwards.

Note that this is only required for submissions to `Submitty` . It is not necessary in labs or on tests; although, if you echo the input back in those situations, you will not be penalized.