MISE Summer Programming Camp

Class 1 Worksheet - My First Programs

Read these instructions carefully: This worksheet has 3 sections. In the first one you will be setting up your programming environment, so you can write and run your own programs. Section 2 contains a series of short answer questions about what we did in class. You might have to run some code in your programming environment to see the result and write it down. Finally, on Section 3 you'll have to write a program and submit it through CodeForces, the platform we mentioned in class.

It should take you around 1 hour to solve all of this and you should try to be done by next Friday(May 5, 2023). If you have questions, post on piazza!

1. Setting up your programming environment

An IDE (Integrated Development Environment) is a software application that provides a comprehensive set of tools for developing and running programs. In the case of Python, an IDE specifically designed for Python programming provides features such as code editing, syntax highlighting, debugging tools, and more. Before you can write your own programs you have to pick and set up an IDE.

Using a browser

In the first class we showed you a web based IDE where you can write and run your own python code in your browser (as a reminder, here is the link to that IDE: https://www.programiz.com/python-programming/online-compiler/). You should try to write and run one or two examples from the first class. Even though this is a fine option to start with, you want to save your code locally and have more control over it and for that you'll want to install a local IDE. We provide some instructions on how to do it on a laptop and then how to do it on a tablet.

Using a laptop

Here is a step-by-step guide on how to install Visual Studio Code, a good IDE for python beginners:

Installing the IDE

1. First download and install the Python interpreter from here: https://www.python.org/downloads/ (it should be version 3.9.2 or more recent). When you first open the installer make sure to click "Add to path"

2. Download and install the most recent version of Visual Studio Code from here: https://code.visualstudio.com/download
3. Before you can write python code, you need to install the python VSCode extension. In VSCode go to the extensions tab (the squares icon on the left) and look up an extension called "Python" and click install. It should be simply called "Python", there are other extensions with Python in their names, but we don't need them for now. You can also install this extension by following this link: https://marketplace.visualstudio.com/items?itemName=ms-python.python

Writing and Running Code

4. Open VSCode and open a new file
5. Click "File" and "Save as' ' and save it as a Python file (there should be a dropdown box with a lot of programming languages, make sure you select Python). The first time you do this you might get some pop ups with some information, you can ignore those
6. Now you need to tell VSCode where the Python interpreter is so that VSCode can find it. There are two ways to do this:
    a. If there's a banner at the bottom that says "Select Python interpreter", you can click on that and then the top-middle of the window should display the file you just installed (and you should click on that)
    b. If you don't see a banner, do Ctrl+Shift+p to pull up the search function at the top-middle and search for Python: Select Interpreter. Once you click on that, it'll display the file you just installed (and you should click on that)
7. Now write your Python code
8. To run your code, right click anywhere on the text area and click on "Run Python File in Terminal". This will open up a small tab with a shell (much like the web IDE) and run your code

To test the above, try creating a file and saving it with the following code:

```
print(“Hello World”)
```

Now run it and make sure the shell shows the message *Hello World*. If you've done all these steps and it's still not working, try restarting your computer. If that still doesn't help then an instructor can try to help you.

This should be it, you should now be able to write and run your own Python code. VSCode has a lot of features you might want to explore. You don't really need any of them to be able to write Python code, but if you are interested in exploring them you can look over here for all the information about VSCode: https://code.visualstudio.com/docs

If you are having any trouble with this, post something on Piazza and we’ll try to help you.

Using a tablet

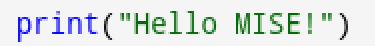
If all you have available to you is a tablet, there are a couple of apps that allow you to program in Python. For ipad users, there is no great free option as far as we know, but there is an app called Pythonista (http://omz-software.com/pythonista/), which we recommend. For android users, we recommend installing an app called Pydroid 3 (https://play.google.com/store/apps/details?id=ru.iiec.pydroid3).
Once you install any of these apps, you should try to run a simple program like one of the ones we wrote in class and see if you can get it to run.

If you are having any trouble with this, post something on Piazza and we'll try to help you.

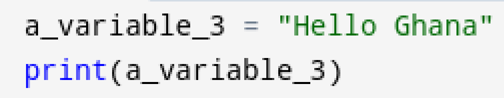
2. Review Questions

The following questions are only review questions, you don't have to submit your answers anywhere or write anything down. They are meant to help you review the topics we discussed in class.

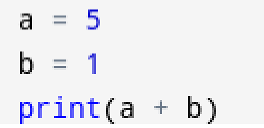
Question 1. The following are short pieces of code similar to what you've seen in class. Look at them first and think about what you think they should do, then type them into your IDE (Visual Studio, one of the tablet apps or the online one) and run them to see if it matches what you thought. If not, try to think about why it does what it does.

a. 

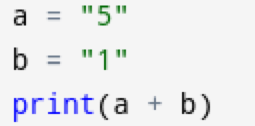
```
print("Hello MISE!")
```

b. 

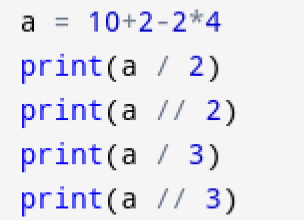
```
a_variable_3 = "Hello Ghana"
print(a_variable_3)
```

c. 

```
a = 5
b = 1
print(a + b)
```

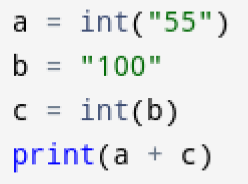
d. 

```
a = "5"
b = "1"
print(a + b)
```

e. 

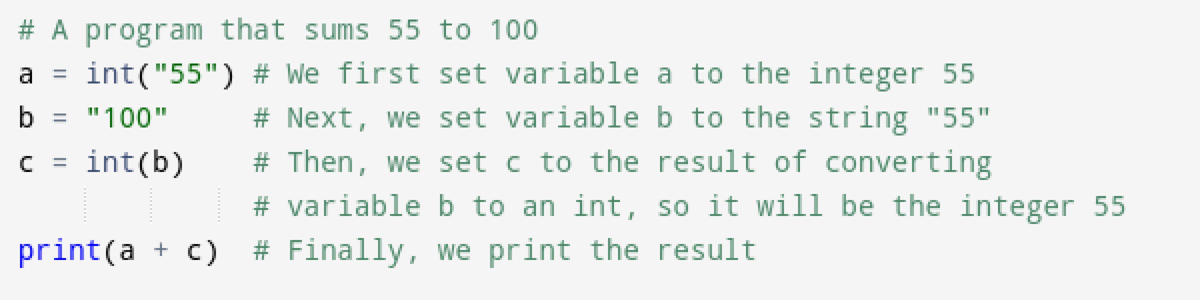
```
a = 10+2-2*4
print(a / 2)
print(a // 2)
print(a / 3)
print(a // 3)
```

f. The *int()* function is another function we saw in class. This function will convert a value of type *string* to a value of type *int*, so it will convert text into numbers. This is a very useful function to know (specially when used with the *input()* function), so here is an example of that:



```
a = int("55")
b = "100"
c = int(b)
print(a + c)
```

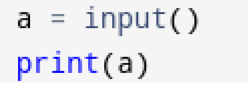
g. In Python a comment is an instruction or piece of code that gets ignored by the compiler/translator. So we can use comments to annotate our code. We use the ‘#’ symbol to indicate a comment and the result is that all the text after one ‘#’ up to the end of the line will be ignored by the compiler. Here is the same code as in the previous paragraph but with comments to help with understanding the code:



```
# A program that sums 55 to 100
a = int("55") # We first set variable a to the integer 55
b = "100"     # Next, we set variable b to the string "55"
c = int(b)    # Then, we set c to the result of converting
              # variable b to an int, so it will be the integer 55
print(a + c)  # Finally, we print the result
```

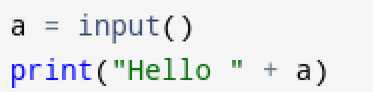
Question 2. Here are some small programming tasks where we need to take some input and perform some operation on it. In each one we need to read some input from the shell and then do something to it. After the description of each task there is a small code that achieves that. Read the task and think about how you’d solve it, then read the solution and see if it matches what you thought. Type up the solution and run it to verify that it does what it’s supposed to do.

a. Write a program that reads a string and write it to the shell.



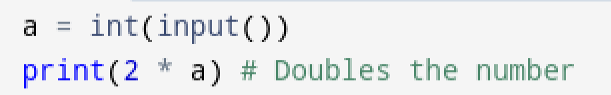
```
a = input()
print(a)
```

b. Write a program that reads a string and writes “Hello” followed by the string to the shell.



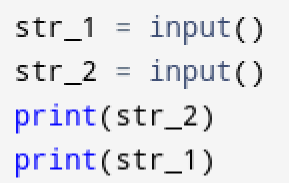
```
a = input()
print("Hello " + a)
```

c. Write a program that reads an integer and writes it’s double to the shell.



```
a = int(input())
print(2 * a) # Doubles the number
```

d. Write a program that reads two strings and writes the second followed by the first to the shell.



```
str_1 = input()
str_2 = input()
print(str_2)
print(str_1)
```

Question 3. Write any program that you think of. Try to play around with the language to see if you can find different things you can do with the functions you learned, even if they cause some error and try to understand why it works (or doesn't if it gives an error). If you find interesting examples, post them to piazza!

3. Coding Questions

The easiest way to learn how to program is by doing. To do so, you'll have to solve tasks that look like the ones from Question 2 of the previous section. To make the tasks less ambiguous, we'll give a lot more detail, but they all require you to read something from the shell (using *input()*), apply some operations to it and then write some result to the shell. To help you with this, we will use a platform called CodeForces (http://codeforces.com/).

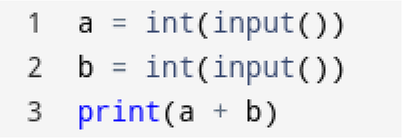
CodeForces is a platform for competitive programming and includes thousands of problems. These are all too hard for now (although maybe by the end of the class you'll be able to solve some of them). **Start by creating an account by going to http://codeforces.com/.** To do so, you should click on "Register" on the top right corner and follow the instructions.

So we can know who you are, **add your name and the codeforces username you selected to the Google Sheet** over here https://docs.google.com/spreadsheets/d/1j4oji3qy1kGWpBBQkjPEHeh18sKWfbOPQOXzg7rNUrk/edit?usp=sharing (do not edit anything else on the Google Sheet).

Now, we are going to use the groups feature of CodeForces. We created a group for this program which you can find herehttps://codeforces.com/group/K1Fxw6skwV/contests. **Go to this link and click on join on the right.**

You should see a contest called "Class 1 Problems". A contest is just a list of problems, there is no competition going on (that's what codeforces names these). **Click on it.** You should see a single problem called "Addition". **Click on it.**

Read the problem statement. In this problem you are supposed to read two numbers, add them and then print their result. **Think about how to do this first and try to write it on your IDE and save it as a .py file**, but since this is the first task, here is one possible solution.



```
a = int(input())
b = int(input())
print(a + b)
```

Now, you can submit your solution to codeforces! To do so, on the right make sure Python 3.9.2 is selected and **choose the program file you made with the solution and click on submit**.

You’ll be redirected to the “My submissions” page where you can see the result of your submission.

CodeForces works by testing your program against multiple test cases that we created. Each test case is a possible input to the program along with a correct output. CodeForces will run your program with each input and compare the result with the output, so it is very important that you don’t print anything else, otherwise it will differ from the expected output. If you wait a little bit, you should see **Accepted** as the verdict of your submission.

We will use CodeForces a lot for the next few classes so make sure you know how to do this. If you have questions about it, post them to piazza. Also, feel free to explore the rest of the group, it will have more content soon!