

## Week 1 Deliverables

**Overview:** In this week, you have set-up your Python Environment. The Lab for this week demonstrates your first use of this environment with a fairly simple Python application. You will also use pylint to verify your code is using professional coding style and standards.

**Submission requirements include 2 files. (Zipping them into one file is acceptable):**

- Python Voter Registration Application Code (python code)
- Word or PDF file containing your test and pylint results

### Python Applications for Lab1: (total 100 points):

This lab consists of two parts. The first exercise produces a voter registration application asking the user a few simple questions followed by a confirmation of registration, provided the user is eligible. The second part documents your testing and pylint analysis results.

**1.** Using your Python programming environment, write a Python application that supports voter registration. The application will launch and run from the command line prompt. The application will prompt the user for their first name, last name, age, country of citizenship, state of residence and zipcode. To be a valid registration all fields must be entered. If they are at least 18 years old and a U.S citizen, they can move forward and be prompted for the remaining questions and register to vote. If not, they should not be presented with the additional questions. There should be some error checking logic on the input statements to make sure the age numbers entered seem reasonable (e.g. a person is probably not > 120 years) and states should be 2 letters representing only valid U.S. States. The application should prompt the user for the needed questions to complete the registration and re-prompt when data is invalid giving the user the opportunity to retry. The output should summarize the input data and congratulate the user if they are eligible to vote and entered all of the data. The user should be given options to exit the program at any time to cancel the registration process.

The following is a possible application interface. Other application interfaces are possible as well. **(80 points)**

```
*****
Welcome to the Python Voter Registration Application.
Do you want to continue with Voter Registration?
Yes.
What is your first name?
Sally
Do you want to continue with the voter Registration?
What is your last name?
Smith
Do you want to continue with the voter Registration?
Yes
What is your age?
49
Do you want to continue with the voter Registration?
```

```

Yes
Are you a U.S. Citizen?
Yes
Do you want to continue with the voter Registration?
Yes
What state do you live?
MD
Do you want to continue with the voter Registration?
Yes
What is your zipcode?
21012
Thanks for registering to vote. Here is the information we
received:
Name (first last): Sally Smith
Age: 49
U.S. Citizen: Yes
State: MD
Zipcode: 21012
Thanks for trying the Voter Registration Application. Your voter
registration card should be shipped within 3 weeks.
*****

```

#### Hints:

1. Be sure to add logic to test for continuing the registration process.
2. Validate data is valid on entry (e.g. all fields have input data, age seems correct, states seem correct.)
3. Test with many combinations. For example, what happens if you enter invalid data? Exit the application at any point, or aren't 18 years old?
4. Use comments to document your code
5. Use pylint to verify the code style – the goal is a 10!

**2. Document your test results for each application within your programming environment. You should also include and discuss your pylint results for each application. The test document should include a test table that includes the input values, the expected results and the actual results. A screen capture should be included that shows the actual test results of running each test case found in the test table. Be sure to include multiple test cases to provide full coverage for all code. For example, you should demonstrate each set of logic in the code works as expected and every statement in the code is reached through the test cases. (20 points)**

A possible test table may look like this:

Test Case	Input	Expected Output	Actual Output	Pass?
1a	1,2,3,4,5,6,7,8,9,10	Sum= 54, Range =9	Sum= 54, Range =9	Yes
1b	2,10,1,11,11,32,42,21,32,23	Sum= 183, Range =41	Sum= 183, Range =41	Yes

1c	3,g,e,s,7,12,e,s,x,!	Error	Error	No input validation issue
...				

Include the screen captures support each of the test cases. These are the screen captures validating the actual output for each test case.

**Any submissions that do not represent work originating from the student will be submitted to the Dean's office and evaluated for possible academic integrity violations and sanctions.**