# Problem Report

|  |  |  |
| --- | --- | --- |
| Assignment: Lab\_GroupA | Date: 09/20/2020 | Report#: 1 |

|  |  |
| --- | --- |
| Testing Team#: Team 10 | Partner Team#: Team 7 |

Testing Team Members:

|  |  |
| --- | --- |
| Mike Horn | Matthew Merle |
| Jerry McNeill | Lee Lewis |

Program: **FullRetirementAge.py, validations.py**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Report Type  (1-3): **2, 3** | 1 - Coding Error  2 - Design Issue  3 - Suggestion | Severity  (1-3): **3** | 1 - Fatal  2 - Serious  3 - Minor | Priority  (1-5): **4** | 1 - Fix IMMEDIATELY  2 - Fix ASAP  3 - Must fix  4 - Fix if possible  5 - Optional |

Problem Summary:

* *normal\_retirement()* should have parameter names that are more specific
* *normal\_retirement()* should be broken up for better readability and debugging
* Separate input validation in *validationYear()* and *validationMonth()* from receiving input from the user
* *age* variable is stored as a String, not an int

Problem Details:

* More detailed parameter names are perfectly fine as it’s not something you’ll be typing – since it’s just describing what’s being passed in, being as detailed as you can with parameter names helps for readability of code
* Although *normal\_retirement()* compiles and runs fine as-is, breaking it into smaller functions will help in the event of any bugs that could arise. Modularized code leads to better readability, and simpler debugging.
* Due to the fact that the *normal\_retirement()* function returns three values, you could run into problems where just one of those values is coming back incorrectly, but you have to work around code that’s working how it should (IE, the two values that return correctly).
* As both *validationYear()* and *validationMonth()* take in the prompt, it makes the code not very reusable. If the project were to expand, that code would have to be re-written due to it being specifically tailored for the prompt, instead of just validating the input.
* Because *age* is stored as a String and not an int in *normal\_retirement()*, it would have to be parsed to be used as an int in the future.

Steps to Reproduce:

* As this isn’t a compile error or any other kind of *true* problem with the code, it isn’t something to be “reproduced”.

Suggested Fix:

* Change parameter names to a more detailed alternative (IE: *year* -> *birthYear*, *month* -> *birthMonth*).
* Separate *normal\_retirement()* function into smaller functions, that only return one value each
* By doing this, if a value is not returned as it should be, you know exactly where to look to fix the problem without even having to debug
* Break up the steps of receiving input and validating input into two separate functions. It’s harder to test as one function instead of two, and you want to be able to reuse your code in the future.
* Store *age* as an int, rather than a String. This’ll allow you to use *age* in the future, instead of having to parse the String to just have the age.