TASK 1. Python Class Generator

A typical python class is defined with the following parameters:

- class name
- super class name
- list of private attributes

For example, if a class has class name as `ClassName`, super class name as `SuperClassName`, and 2 private attributes: `attr1` and `attr2`. We could follow the following format to generate a standard python class:

```
class ClassName(SuperClassName):                Class definition
    def __init__(self, attr1, attr2):           __init__ function
        self._attr1 = attr1
        self._attr2 = attr2

    def set_attr1(self, new_attr1):             Mutators
        self._attr1 = new_attr1

    def set_attr2(self, new_attr2):
        self._attr2 = new_attr2

    def get_attr1(self):                        Accessors
        return self._attr1

    def get_attr2(self):
        return self._attr2

    def __str__(self):                          __str__ function
        result = ""

        return result
```

Write a class named `PythonClass`, which should take in 3 **private** attributes:

c_name	string
sc_name	string
attr_list	list of string

The class should contain at least the following 2 methods:

`__init__()`	Initializer for the class
`__str__()`	Generate a long string which contains the class definition, __init__function, mutators, accessors and the dummy __str__ function.

Task 1.1

- Write program code for `PythonClass`
- Test your code by using:
```
pc = PythonClass("Person", "object", ["name", "age"])
print(pc)
```

Create a class named `PythonClassGenerator` or `PCG`.

The class contains 1 **private** attribute, `pc_list`, and it should be assigned as an empty list upon initializing.

The class contains 2 public methods:

`add_pc()`	Add a `PythonClass` object to the `pc_list`
`read_file()`	This function takes in a file name, and process the file content line by line, and create `PythonClass` objects to be added into the `pc_list`. Each line in the source file are formatted according to the following format: `[class name]; [super class name]; [list of attributes separated by ","]`
`generate()`	This function will open a file named "`new_oop.py`" and generate all python classes stored in the `pc_list` to this file.
`display()`	Print out the class name of the `PythonClass` objects inside the current `pc_list`.

Task 1.2

- Write program code for `PythonClassGenerator` or `PCG`.
- Test your code by using:
  `new_class.txt`

Create a text-based user interface with the following options:

```
1. Get User Input
2. Display
3. Read File
4. Generate OOP File and End
```

Task 1.3

- Write program code for the menu with relevant data validations.