## Programming Project

Create a Python project for the following problem. Zip the Python project into a zip file. Submit the zip file to Blackboard for credit.

You will write a python program for customers of a store to create acounts. Each customer needs to provide first name, last name, age, email address, desired password, credit card number and three-digit security code. The program will create two customer accounts and store their data in a text file.

The following lists the requirements of this project.

### Instructions – Customer Module

Define a Customer class in this module (file name: customer.py). The following UML diagram shows the design of this class:

|  |
| --- |
| **Customer** |
| - last\_name: String  - first\_name: String  - age: Integer  - email: String  - password: String  - card\_number: String  - security\_code: String |
| + \_\_init\_\_(email: String)  + input\_age()  + input\_password()  + input\_card\_number()  + input\_security\_code()  + input\_info()  + verify\_info()  + output\_info(): String |

This class has seven private instance variables to store customer’s personal information: first\_name, last\_name, age, email, password, card\_number and security\_code.

The class has eight public methods:

1. \_\_init\_\_ – The constructor receives email address and stores it in the instance variable email. It also creates the other instance variables and initializes them. Initialize age to 0. Initialize the other instance variables to empty strings.
2. Input\_age – In this method, the user enters age. Age is invalid if it is negative or if the input string cannot be converted to an integer. If an invalid age is entered, display an error message and ask the user to reenter. Repeat this until the age is valid. Store the valid age in the instance variable age. This method has no parameter (except self) and no return value.
3. Input\_password – In this method, the user enters a password meeting the password requirements. The password must be 8-12 characters in length, and must contain at least one upper-case letter, one lower-case letter, and one number. If an invalid password is entered, display an error message and ask the user to reenter. Repeat this until the password is valid. Store the valid password in the instance variable password. This method has no parameter (except self) and no return value.
4. input\_card\_number – In this method, the user enters a 16-digit credit card number. The card number is invalid if the length is not 16 or if it has one or more non-digit characters. If an invalid card number is entered, display an error message and ask the user to reenter. Repeat this until the card number is valid. Store the valid card number in the instance variable card\_number. This method has no parameter (except self) and no return value.
5. input\_security\_code – In this method, the user enters a 3-digit credit card security code. The security code is invalid if the length is not 3 or if it has one or more non-digit characters. If an invalid security code is entered, display an error message and ask the user to reenter. Repeat this until the security code is valid. Store the valid security code in the instance variable security\_code. This method has no parameter (except self) and no return value.
6. input\_info – This method allows the user to enter first name, last name and other personal information. First, ask the user to enter first name and last name and store them in the instance variables first\_name and last\_name. Then call the following methods to enter other personal information: input\_age, input\_password, input\_card\_number and input\_security\_code. This method has no parameter (except self) and no return value.
7. verify\_info – This method allows the user to update any item previously entered before completing the registration. First, display all instance variables. Then ask the user to choose one item to update. Write code or call appropriate method to update the chosen item. Repeat this until the user does not want to make any more changes. This method has no parameter (except self) and no return value.
8. output\_info – This method creates and returns a string that consists of all personal information of the customer. Data in the string must follow this order: first\_name, last\_name, age, email, password, card\_number and security\_code. Insert a space after each item except the security\_code. Insert a newline character, i.e. ‘\n’, at the end of the string. The returned string will be used later to write customer’s personal information to a textfile. This method has no parameter (except self).

### Instructions – Main Module

The main module of the project must do the following:

1. Create two Customer objects. Since you need to pass email address to the \_\_init\_\_ method when you create a Customer object, ask the user to enter the email address of the customer first.
2. For each Customer object, call the input\_info method to input personal information and call the verify\_info method to verify peronal information.
3. Write personal information of the two customers to the textfile ‘customers.txt’. Each customer will occupy one line. Personal information of each customer must be written in this order: first\_name, last\_name, age, email, password, card\_number and security\_code. Insert a space after each item except the security\_code. Insert a newline character, i.e. ‘\n’, at the end of the line.

### Sample Input/Output

Customer 1

Enter email address: atkinson@waketech

First name: Ali

Last name: Atkinson

Enter age: 24.5

Error: Age must be an integer.

Enter age: 24

Your password must be 8-12 characters long containing at least one upper-case letter, one lower-case letter, and one number.

Enter password: 12345678

does not contain upper-case

Your password must be 8-12 characters long containing at least one upper-case letter, one lower-case letter, and one number.

Enter password: 1234567X

does not contain lower-case

Your password must be 8-12 characters long containing at least one upper-case letter, one lower-case letter, and one number.

Enter password: abcdEFGH

does not contain number

Your password must be 8-12 characters long containing at least one upper-case letter, one lower-case letter, and one number.

Enter password: 123456Xg

Valid Password.

Enter 16-digit credit card number: 1234

Card number must be 16 digits.

Enter 16-digit credit card number: 12345678abcdefgh

Card number must be 16 digits.

Enter 16-digit credit card number: 1234567812345678

Enter 3-digit security code: abc

PIN must be three digits.

Enter 3-digit security code: 1234

PIN must be three digits.

Enter 3-digit security code: 123

1. First name: Ali

2. Last name: Atkinson

3. Email address: atkinson@waketech

4. Password: 123456Xg

5. Age: 24

6. Card number: 1234567812345678

7. Security code: 123

To correct any entry, enter the entry's number and press RETURN. If everything is correct, press 0: 3

Email address: [atkinson@waketech.edu](mailto:atkinson@waketech.edu)

1. First name: Ali

2. Last name: Atkinson

3. Email address: atkinson@waketech.edu

4. Password: 123456Xg

5. Age: 24

6. Card number: 1234567812345678

7. Security code: 123

To correct any entry, enter the entry's number and press RETURN. If everything is correct, press 0: 5

Enter age: 25

1. First name: Ali

2. Last name: Atkinson

3. Email address: atkinson@waketech.edu

4. Password: 123456Xg

5. Age: 25

6. Card number: 1234567812345678

7. Security code: 123

To correct any entry, enter the entry's number and press RETURN. If everything is correct, press 0: 4

Your password must be 8-12 characters long containing at least one upper-case letter, one lower-case letter, and one number.

Enter password: XX1234gg

Valid Password.

1. First name: Ali

2. Last name: Atkinson

3. Email address: atkinson@waketech.edu

4. Password: XX1234gg

5. Age: 25

6. Card number: 1234567812345678

7. Security code: 123

To correct any entry, enter the entry's number and press RETURN. If everything is correct, press 0: 7

Enter 3-digit security code: 9876

PIN must be three digits.

Enter 3-digit security code: 987

1. First name: Ali

2. Last name: Atkinson

3. Email address: atkinson@waketech.edu

4. Password: XX1234gg

5. Age: 25

6. Card number: 1234567812345678

7. Security code: 987

To correct any entry, enter the entry's number and press RETURN. If everything is correct, press 0: 0

Registration and verification completed for this customer.

Customer 2

Enter email address: bbenson@waketech.edu

First name: Bob

Last name: Benson

Enter age: 19

Your password must be 8-12 characters long containing at least one upper-case letter, one lower-case letter, and one number.

Enter password: AAss1256

Valid Password.

Enter 16-digit credit card number: 4455778811229900

Enter 3-digit security code: 456

1. First name: Bob

2. Last name: Benson

3. Email address: bbenson@waketech.edu

4. Password: AAss1256

5. Age: 19

6. Card number: 4455778811229900

7. Security code: 456

To correct any entry, enter the entry's number and press RETURN. If everything is correct, press 0: 0

Registration and verification completed for this customer.

Data of two customers written to the file 'customers.txt'.

### Grading

Zip your Python project and submit it to Blackboard for credit. Your Python project should have at least two files: a Python file containing the definition of Customer class, and a Python file containing the main module.

\_\_init\_\_ method of Customer class [10 points]

input\_age method of Customer class [10 points]

input\_password method of Customer class [10 points]

input\_card\_number method of Customer class [10 points]

input\_security\_code method of Customer class [10 points]

input\_info method of Customer class [10 points]

verify\_info method of Customer class [10 points]

output\_info method of Customer class [10 points

Creating Customer objects and calling their methods [10 points]

Write data of Customer objects to text file [10 points]