



EGR680 High Level Implementation on FPGA

Laboratory 07

Python Functions, File I/O, and Object-Oriented Programming

Professor: Dr. C. Parikh

Student: Dimitri Häring

October 24, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Design</b>	<b>3</b>
2.1	ATM machine Part I . . . . .	3
2.2	ATM machine Part II . . . . .	3
2.3	ATM machine Part III . . . . .	4
<b>3</b>	<b>Conclusion</b>	<b>6</b>
<b>4</b>	<b>Appendix</b>	<b>7</b>
4.1	Python code Part I . . . . .	7
4.2	Python code Part II . . . . .	11
4.3	Python code Part III . . . . .	15

# 1 Introduction

The goal of laboratory seven is to familiarize the student with functions, file I/O, and objective oriented programming in python.

## 2 Design

In this section the design and decisions that were made to achieve the laboratory are discussed.

### 2.1 ATM machine Part I

The software of the previous lab has been split up into two files so that the main logic and main program flow is in the main file and the functions are separated in an atm\_func file. Furthermore a file I/O for login errors was implemented to aid assistance by unusual behavior. The two code listings are shown in Listing 8 and 9. A single function is shown in Listing 1.

```
1 def mainMenu ( ) :  
2     print ( " Main Menu" )  
3     print ( " _____ \n" )  
4     print ( " 1 . CHECK BALANCE" )  
5     print ( " 2 . WITHDRAW CASH" )  
6     print ( " 3 . DEPOSIT CASH" )  
7     print ( " 4 . CHANGE PIN" )  
8     print ( " 5 . EXIT " )  
9     return 1 # no error 1 , error 0
```

Listing 1: Python function mainMenu .

The following Listing shows the function call in the main program. That shows how bigger programs in python can be structured sequentially.

```
1 mainMenu ( )
```

Listing 2: Python function call mainMenu .

The following listing shows the error log output file content. This can be easily used to check which errors has been tested and has a message generated to it. The error log is continuously appended into a text file and has to be managed manually so far this can be automated with a script as example.

Listing 3: Error log Part II.

```
Thu Oct 25 09:15:45 2018ATM program starts  
Thu Oct 25 09:15:54 2018 User logged in  
Thu Oct 25 09:16:06 2018 Withdraw error  
Thu Oct 25 09:16:19 2018 Deposit error  
Thu Oct 25 09:16:34 2018 User PIN error  
Thu Oct 25 09:27:05 2018 Program Closed
```

### 2.2 ATM machine Part II

In this part the program was expanded with an receipt function that would log all the transactions of a ongoing session and let the user decide to print it out the end of the session. Figure 1 shows how the user can decide after exiting the program if he wishes to print the receipt or not. The python code himself for file I/O used can be examined in closer detail in Listing 10 and 11. First a file has to be open this can be done with open(<filename>, mode) and then be accessed with the functions .read() and .write().

```


Main Menu
-----

1. CHECK BALANCE
2. WITHDRAW CASH
3. DEPOSIT CASH
4. CHANGE PIN
5. EXIT

PLEASE SELECT FROM THE ABOVE OTPTIONS: 5

Do you wish to print the receipt, press y or n? y

-----Bank Me-----
-----Transaction Receipt-----
*****
Your checking account balance is----> $0
Your withdraw amounte is-----> $0
Your deposit amounte is-----> $3
Your checking account balance is----> $3
You have changed your PIN number
Your withdraw amounte is-----> $1
Your checking account balance is----> $2
THANK YOU FOR CHOOSING BANK ME!

In [49]:  printed receipt

```

Figure 1: Printed receipt.

Listening 4 shows how a text file can be opened either with append mode or write mode. It shows also how errors can be handled with try and except commands. The lowest line show how to close a file.

```

1 # Open a file
2 try:
3     fo = open("07_lab_error_log.txt", "a")
4     fr = open(atm.ReceiptFile, "w")
5     fo.write(asctime()+ ' ATM program starts ' + '\n')
6     fr.write('\n-----Bank Me-----\n')
7     fr.write('-----Transaction Receipt-----\n')
8     fr.write('*****\n\n')
9 except IOError as e:
10     print('File '+e.filename+' could not be opened or written to!')
11
12 fo.write(asctime()+ ' Program Closed\n')
13 fo.close() # to ensure file is closed
14 fr.close() # to ensure file is closed

```

Listing 4: Python file I/O example.

## 2.3 ATM machine Part III

This section shall demonstrated the objective programming that allows two crate a class an instance this class as instances or an instance. A class contains usually data objects and methods. Listening 5 shows the class named Atm with class variables followed by the `__init__` function which is first call after instantiation of a class and serves to initializes specific variables. next the normal statements or so called methods are

listed which can be used in the main program. Notice, the word *self* which is used in a method which defines a method and separate it from a normal static function.

```
1 import re
2 class Atm:
3     AMOUNT_MIN = 0
4     AMOUNT_MAX = 1000
5
6     Pin = "1234"
7     Select = 0
8     AccountValue = 0
9     ReceiptFile = "atm_receipt.txt"
10
11     def __init__(self, AccountValue, Pin):
12         self.AccountValue = AccountValue
13         self.Pin = Pin
14
15     # Print Receipt
16     def receipt(self, filename, file):
17         file.write('THANK YOU FOR CHOOSING BANK ME! ')
18         file.close()
19         file = open(filename, "r")
20         for line in file:
21             print(line, end='')
22         file.close()
23     return 1
```

Listing 5: Python class example with a method.

Listening 6 shows how a class can be imported with from import command and how it can be instantiated with initial variables as an object called atm. It also shows how methods can be used as example atm.welcome(). Class variables can be accessed like atm.Pin as example.

```
1 from atm_class import Atm
2 atm = Atm(100, '2345') # instantiate class Atm to atm object
3
4 # Star program
5 if atm.welcome(): # first method used from the atm object
6     error = 0
7 else:
8     error = 1
9     fo.write(asctime()+ ' welcome() error\n')
10
11 # PIN validation
12 while True:
13     ret = 0
14     if atm.pin( input("Please enter your PIN: "), atm.Pin):
15         error = 0
16         fo.write(asctime()+ ' User logged in \n')
17         break
18     else:
19         error = 1
20         fo.write(asctime()+ ' User login error\n')
```

Listing 6: Python object example with a method.

The following listening shows the error log output file content. This can be easily used to check which errors has been tested and has a message generated to it. The error log is continuously appended into a text file and has to be managed manually so far this can be automated with a script as example.

Listing 7: Error log Part III.

```
Thu Oct 25 15:53:33 2018 ATM program starts
Thu Oct 25 15:53:37 2018 User login error
Thu Oct 25 15:53:40 2018 User login error
Thu Oct 25 15:53:44 2018 User login error
```

```

Thu Oct 25 15:53:46 2018 User login error
Thu Oct 25 15:53:48 2018 User logged in
Thu Oct 25 15:54:13 2018 Deposit error
Thu Oct 25 15:54:26 2018 Deposit error
Thu Oct 25 15:54:54 2018 Withdraw error
Thu Oct 25 15:55:00 2018 Withdraw error
Thu Oct 25 15:56:47 2018 Program Closed

```

Figure 2 shows the receipt output generated by the program written in part three where a class of an ATM was instantiated instead of functions methods where used. This together with the error log confirms the correct functioning of the ATM.

```

Main Menu
-----

1. CHECK BALANCE
2. WITHDRAW CASH
3. DEPOSIT CASH
4. CHANGE PIN
5. EXIT

PLEASE SELECT FROM THE ABOVE OTPTIONS: 5

Do you wish to print the receipt, press y or n? y

-----Bank Me-----
-----Transaction Receipt-----
*****
Your checking account balance is----> $100
Your deposit amounte is-----> $0
Your deposit amounte is-----> $45
Your checking account balance is----> $145
Your withdraw amounte is-----> $0
You have changed your PIN number
THANK YOU FOR CHOOSING BANK ME! printed receipt

```

Figure 2: Printed receipt Part III.

Further work would be to write the file I/O access in better methods which makes it nicer to look and easier to handle as well as the error log handling is still plumb and needs a better strategy if the program would be advanced.

### 3 Conclusion

The lab demonstrates the use of the python as simple and fast scripting language that allows access to vast number of packages that allows an decreased development time. The syntax is easy to learn but it is possible to lose the overview by having too many continuations of statements. Furthermore, it allows file I/O to save data on a disk and objective oriented programming with classes.

## 4 Appendix

The appendix contains code listening and other large information parts that contain partial or complete relevance to the reports topic.

### 4.1 Python code Part I

```
1  -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4
5  This is a temporary script file.
6  """
7  from time import asctime
8
9  from atm_func_partI import welcome, pin, \
10 selMainMenue, newPin, deposit, withdraw, balance
11
12 AMOUNT_MIN = 0
13 AMOUNT_MAX = 1000
14
15 gPin = "1234"
16 gSelect = 0
17 gAccountValue = 0
18
19 # Open a file
20 try:
21     fo = open("07_lab_error_log.txt", "a")
22     fo.write(asctime()+ 'ATM program starts ' + '\n')
23 except IOError as e:
24     print('File '+e.filename+' could not be opened!')
25
26 # Star program
27 if welcome():
28     error = 0
29 else:
30     error = 1
31     fo.write(asctime()+ ' welcome() error\n')
32
33 # PIN validation
34 while True:
35     ret = 0
36     if pin( input("Please enter your PIN: "), gPin):
37         error = 0
38         fo.write(asctime()+ ' User logged in \n')
39         break
40     else:
41         error = 1
42         fo.write(asctime()+ ' User login error\n')
43
44 # Main menu
45 while gSelect != "5":
46     ret = 0
47     ret = selMainMenue(input("PLEASE SELECT FROM THE ABOVE OTPTIONS: ") )
48     # print(ret) # debug only
49     if (ret[0]):
50         error = 0
51         gSelect = ret[1]
52     # print(ret[1] ) # debug only
53     else: # error
54         error = 1
55         gSelect = 0
56         fo.write(asctime()+ ' User slection error\n')
57         print( ret[0])
58
59     if gSelect == '5': # exit
```

```

60     exit
61     elif gSelect == '4': # change PIN
62         while True:
63             ret = 0
64             ret = newPin(input("ENTER YOUR NEW PIN: "))
65             # print(ret) # debug only
66             if (ret[0]):
67                 error = 0
68                 gPin = ret[1]
69                 # print(ret[1] ) # debug only
70                 break
71             else: # error
72                 error = 1
73                 gSelect = 0
74                 fo.write(asctime()+ ' User PIN error\n')
75                 break
76
77     elif gSelect == '3': # Deposit
78         while True:
79             ret = 0
80             ret = deposit(input("ENTER YOUR DEPOSIT AMOUNT IN $: ") \
81                             , AMOUNT_MIN, AMOUNT_MAX, gAccountValue)
82             # print(ret) # debug only
83             if (ret[0]):
84                 error = 0
85                 gAccountValue = ret[1]
86                 # print(ret[1] ) # debug only
87                 break
88             else: # error
89                 error = 1
90                 gSelect = 0
91                 fo.write(asctime()+ ' Deposit error\n')
92
93     elif gSelect == '2': # Withdraw
94         while True:
95             ret = 0
96             ret = withdraw(input("ENTER YOUR WITHDRAW AMOUNT IN $: ") \
97                             , AMOUNT_MIN, AMOUNT_MAX, gAccountValue)
98             # print(ret) # debug only
99             if (ret[0]):
100                 error = 0
101                 gAccountValue = ret[1]
102                 # print(ret[1] ) # debug only
103                 break
104             else: # error
105                 error = 1
106                 gSelect = 0
107                 fo.write(asctime()+ ' Withdraw error\n')
108
109     elif gSelect == '1': # Balance
110         balance(gAccountValue)
111     else:
112         print("\nIncorrect Seleccion!")
113
114     print("\nThank you for chosing BANK ME!")
115     fo.write(asctime()+ ' Program Closed\n')
116     fo.close()
117     fr.close()

```

Listing 8: Python code for an ATM Part I.

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Oct 18 18:34:49 2018
4
5  @author: schwa
6  """
7  import re

```



```

8 # Balance
9 def balance( account_value , file):
10     print("\nYOUR BALANCE IS $%.2f \n \n\r" % account_value)
11     file.write('Your checking account balance is————> $'\
12             +str(account_value)+'\n')
13     delay(3000)
14     mainMenue( )
15     return 1
16 # Withdraw money
17 def withdraw( inVal, AMOUNT_MIN, AMOUNT_MAX, account_value, file ):
18     ret = [0, 0]; # no error 1, error 0
19     amount = 0
20     if not re.match("[0-9]*$", inVal):
21         amount = 0
22         print("\nInvalid Input only numbers are allowed.\n")
23         delay(3000)
24         mainMenue( )
25         ret=[0, inVal]
26     # break
27     else:
28         amount = int(inVal)
29         if amount >= AMOUNT_MIN and amount <= account_value :
30             account_value = account_value - amount
31             print("\nWITHDRAW AMOUNT $%.2f\n" % amount)
32             file.write('Your withdraw amounte is————> $'\
33                     +str(amount)+'\n')
34             amount = 0
35             delay(3000)
36             mainMenue( )
37             ret=[1, account_value]
38     # break
39     else:
40         ret=[0, inVal]
41         print("\nWITHDRAW AMOUNT $%.2f TOO BIG OR TOO SMALL\n" % amount)
42     return ret
43 # Deposit money
44 def deposit( inVal, AMOUNT_MIN, AMOUNT_MAX, account_value, file ):
45     ret = [0, 0]; # no error 1, error 0
46     amount = 0
47     if not re.match("[0-9]*$", inVal):
48         amount = 0
49         print("\nInvalid Input only numbers are allowed.\n")
50         delay(3000)
51         mainMenue( )
52         ret=[0, inVal]
53     else:
54         amount = int(inVal)
55         if amount >= AMOUNT_MIN and amount <= AMOUNT_MAX :
56             account_value = account_value + amount
57             print("\nDEPOSIT AMOUNT $%.2f\n" % amount)
58             file.write('Your deposit amounte is————> $'\
59                     +str(amount)+'\n')
60             amount = 0
61             delay(3000)
62             mainMenue( )
63             ret=[1, account_value]
64     else:
65         ret=[0, inVal]
66         print("\nDEPOSIT AMOUNT $%.2f TOO BIG OR TOO SMALL\n" % amount)
67     return ret
68 # set new pin
69 def newPin( inVal, file ):
70     ret = [0, 0]; # no error 1, error 0
71     if not re.match("[0-9]{4}$", inVal):
72         print ("Error! Make sure you only use numbers from 0–9 in PIN")
73         inVal = 'Z'
74         ret=[0, inVal]
75     else:

```

```

76         ret =[1, inVal]
77     #         print(ret) # debug only
78     print("\nYOUR NEW PIN IS", inVal, "\n")
79     file.write('You have changed your PIN number\n')
80     delay(3000)
81     mainMenue()
82     return ret
83 # selection main menue
84 def selMainMenue( inVal ):
85     ret = 0; # no error 1, error 0
86     # inVal = input("PLEASE SELECT FROM THE ABOVE OTPTIONS: ")
87     if not re.match("^[1-5]*$", inVal):
88         print ("Error! Make sure you only use numbers from 1-5 in selecion")
89         ret=[0, inVal]
90     else:
91         ret =[1, inVal]
92     return ret
93 # PIN validation
94 def pin( inVal, PIN ):
95     ret = 0; # no error 1, error 0
96     # while True:
97     #     inVal = input("Please enter your PIN: ") # regex
98     if not re.match("[0-9]{4}$", inVal):
99         print ("Error! Make sure you only use numbers from 0-9 in PIN")
100        inVal = 'Z'
101        ret = 0
102    else:
103        if inVal == PIN :
104            #print("\nCorect PIN") # debug only
105            print ("\n")
106            mainMenue()
107            inVal = 'Z'
108            ret = 1
109        #         break
110        else:
111            print ("\nInvalid PIN!")
112            inVal = 'Z'
113            ret = 0
114    return ret
115
116 def delay( msec ):
117     cnt = 0
118     while cnt < msec:
119         cnt += 0.0001
120     return 1 # no error 1, error 0
121
122 def mainMenue( ):
123     print(" Main Menu")
124     print("----- \n")
125     print(" 1. CHECK BALANCE")
126     print(" 2. WITHDRAW CASH")
127     print(" 3. DEPOSIT CASH")
128     print(" 4. CHANGE PIN")
129     print(" 5. EXIT ")
130     return 1 # no error 1, error 0
131
132 def welcome( ):
133     print("\n      $$$$$$          $$          $$$          $$  $$  $  ")
134     print("      $      $          $$$$      $ $ $          $ $  $ $  $  ")
135     print("      $      $          $  $      $ $ $          $ $  $ $  $  ")
136     print("      $      $          $ $  $      $ $  $          $ $  $  $  ")
137     print("      $      $          $ $  $      $ $  $          $ $  $  $  ")
138     print("      $$$$$$          $ $  $      $ $  $          $ $  $  $  ")
139     print("      $$$$$$          $$$$$$$$$$      $ $  $          $ $  $  ")
140     print("      $      $          $ $          $ $          $ $  $  $  ")
141     print("      $      $          $ $          $ $          $ $  $  $  ")
142     print("      $      $          $ $          $ $          $ $  $  $  ")
143     print("      $      $          $ $          $ $          $ $  $  $  ")

```

```

144 print ("          $$$$      $ $      $ $      $ $      $ $      $ $      $ \n")
145 print ("          $$$$      $ $      $ $      $ $ $ $ $ $ $ $ $ $ $ $      ")
146 print ("          $ $ $      $ $      $ $      $ $ $ $ $ $ $ $ $ $ $ $      ")
147 print ("          $ $ $      $ $      $ $      $ $      $ $      $ $      ")
148 print ("          $ $ $      $ $      $ $      $ $      $ $      $ $      ")
149 print ("          $ $      $ $      $ $      $ $      $ $      $ $      ")
150 print ("          $ $      $ $      $ $      $ $ $ $ $ $ $ $ $ $ $ $      ")
151 print ("          $ $      $ $      $ $      $ $ $ $ $ $ $ $ $ $ $ $      ")
152 print ("          $ $      $ $      $ $      $ $      $ $      $ $      ")
153 print ("          $ $      $ $      $ $      $ $      $ $      $ $      ")
154 print ("          $ $      $ $      $ $      $ $      $ $      $ $      ")
155 print ("          $ $      $ $      $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $      ")
156 print ("          $ $      $ $      $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $      ")
157 print ("_____")
158 print ("                          Welcome to Bank Me ")
159 print ("_____")
160 return 1 # no error 1, error 0

```

Listing 9: Python functions for an ATM Part I.

## 4.2 Python code Part II

```

1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7 from time import asctime
8 import re
9 from atm_func_partII import welcome, pin, \
10 selMainMenu, newPin, deposit, withdraw, balance, receipt, delay
11
12 AMOUNT_MIN = 0
13 AMOUNT_MAX = 1000
14
15 gPin = "1234"
16 gSelect = 0
17 gAccountValue = 0
18 gReceiptFile = "atm_receipt.txt"
19
20 # Open a file
21 try:
22     fo = open("07_lab_error_log.txt", "a")
23     fr = open(gReceiptFile, "w")
24     fo.write(asctime()+ 'ATM program starts ' + '\n')
25     fr.write('\n-----Bank Me-----\n')
26     fr.write('\n-----Transaction Receipt-----\n')
27     fr.write('*****\n\n')
28 except IOError as e:
29     print('File '+e.filename+' could not be opened!')
30
31 # Star program
32 if welcome():
33     error = 0
34 else:
35     error = 1
36     fo.write(asctime()+ ' welcome() error\n')
37
38 # PIN validation
39 while True:
40     ret = 0
41     if pin( input("Please enter your PIN: "), gPin):
42         error = 0
43         fo.write(asctime()+ ' User logged in \n')
44         break
45     else:
46         error = 1

```

```

47         fo.write(asctime()+ ' User login error\n')
48
49 # Main menu
50 while gSelect != "5":
51     ret = 0
52     ret = selMainMenu(input("PLEASE SELECT FROM THE ABOVE OTPTIONS: ") )
53     # print(ret) # debug only
54     if (ret[0]):
55         error = 0
56         gSelect = ret[1]
57     # print(ret[1] ) # debug only
58     else: # error
59         error = 1
60         gSelect = 0
61         fo.write(asctime()+ ' User slection error\n')
62         print(ret[0])
63
64     if gSelect == '5': # exit
65         while True:
66             if not re.match("^[y]$", input('Do you wish to print the receipt , press y or n?
67             ')):
68                 print("\nThank you for chosing BANK ME!")
69                 break
70             else:
71                 receipt(gReceiptFile , fr)
72                 break
73         exit
74     elif gSelect == '4': # change PIN
75         while True:
76             ret = 0
77             ret = newPin(input("ENTER YOUR NEW PIN: ") , fr)
78             # print(ret) # debug only
79             if (ret[0]):
80                 error = 0
81                 gPin = ret[1]
82                 # print(ret[1] ) # debug only
83                 break
84             else: # error
85                 error = 1
86                 gSelect = 0
87                 fo.write(asctime()+ ' User PIN error\n')
88                 break
89
90     elif gSelect == '3': # Deposit
91         while True:
92             ret = 0
93             ret = deposit(input("ENTER YOUR DEPOSIT AMOUNT IN $: ") \
94             , AMOUNT_MIN, AMOUNT_MAX, gAccountValue, fr)
95             # print(ret) # debug only
96             if (ret[0]):
97                 error = 0
98                 gAccountValue = ret[1]
99                 # print(ret[1] ) # debug only
100                 break
101             else: # error
102                 error = 1
103                 gSelect = 0
104                 fo.write(asctime()+ ' Deposit error\n')
105
106     elif gSelect == '2': # Withdraw
107         while True:
108             ret = 0
109             ret = withdraw(input("ENTER YOUR WITHDRAW AMOUNT IN $: ") \
110             , AMOUNT_MIN, AMOUNT_MAX, gAccountValue, fr)
111             # print(ret) # debug only
112             if (ret[0]):
113                 error = 0
114                 gAccountValue = ret[1]

```

```

114         #         print(ret[1] ) # debug only
115         break
116     else: # error
117         error = 1
118         gSelect = 0
119         fo.write(asctime()+ ' Withdraw error\n')
120
121     elif gSelect == '1': # Balance
122         balance(gAccountValue, fr)
123     else:
124         print("\nIncorrect Seleccion!")
125
126 fo.write(asctime()+ ' Program Closed\n')
127 fo.close() # to ensure file is closed
128 fr.close() # to ensure file is closed
129 delay(3000)

```

Listing 10: Python code for an ATM Part II.

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Oct 18 18:34:49 2018
4
5  @author: schwa
6  """
7  import re
8  # Print Receipt
9  def receipt(filename, file):
10     file.write('THANK YOU FOR CHOOSING BANK ME! ')
11     file.close()
12     file = open(filename, "r")
13     for line in file:
14         print(line, end='')
15     file.close()
16     return 1
17 # Balance
18 def balance( account_value , file):
19     print("\nYOUR BALANCE IS $%.2f \n \n\r" % account_value)
20     file.write('Your checking account balance is——> $'\
21             +str(account_value)+'\n\n')
22     delay(3000)
23     mainMenue( )
24     return 1
25 # Withdraw money
26 def withdraw( inVal, AMOUNT_MIN, AMOUNT_MAX, account_value, file ):
27     ret = [0, 0]; # no error 1, error 0
28     amount = 0
29     if not re.match("^[0-9]*$", inVal):
30         amount = 0
31         print("\nInvalid Input only nmbers are alowed.\n")
32         ret=[0, inVal]
33     else:
34         amount = int(inVal)
35         if amount >= AMOUNT_MIN and amount <= account_value :
36             account_value = account_value - amount
37             print("\nWITHDRAW AMOUNT $%.2f\n" % amount)
38             file.write('Your withdraw amounte is——> $'\
39                     +str(amount)+'\n\n')
40             amount = 0
41             delay(3000)
42             mainMenue( )
43             ret=[1, account_value]
44         else:
45             ret=[0, inVal]
46             print("\nWITHDRAW AMOUNT $%.2f TOO BIG OR TOO SMALL\n" % amount)
47     return ret
48 # Deposit money
49 def deposit( inVal, AMOUNT_MIN, AMOUNT_MAX, account_value, file ):

```

```

50     ret = [0, 0]; # no error 1, error 0
51     amount = 0
52     if not re.match("[0-9]*$", inVal):
53         amount = 0
54         print("\nInvalid Input only numbers are allowed.\n")
55         ret=[0, inVal]
56     else:
57         amount = int(inVal)
58         if amount >= AMOUNT_MIN and amount <= AMOUNT_MAX :
59             account_value = account_value + amount
60             print("\nDEPOSIT AMOUNT $%.2f\n" % amount)
61             file.write('Your deposit amounte is—————> $'\
62                     +str(amount)+'\n\n')
63             amount = 0
64             delay(3000)
65             mainMenue( )
66             ret=[1, account_value]
67         else:
68             ret=[0, inVal]
69             print("\nDEPOSIT AMOUNT $%.2f TOO BIG OR TOO SMALL\n" % amount)
70     return ret
71 # set new pin
72 def newPin( inVal, file ):
73     ret = [0, 0]; # no error 1, error 0
74     if not re.match("[0-9]{4}$", inVal):
75         print ("Error! Make sure you only use numbers from 0–9 in PIN")
76         delay(3000)
77         mainMenue( )
78         inVal = 'Z'
79         ret=[0, inVal]
80     else:
81         ret =[1, inVal]
82     # print(ret) # debug only
83     print("\nYOUR NEW PIN IS", inVal, "\n")
84     file.write('You have changed your PIN number\n\n')
85     delay(3000)
86     mainMenue()
87     return ret
88 # selection main menue
89 def selMainMenue( inVal ):
90     ret = 0; # no error 1, error 0
91     if not re.match("[1-5]*$", inVal):
92         print ("Error! Make sure you only use numbers from 1–5 in selecion")
93         ret=[0, inVal]
94     else:
95         ret =[1, inVal]
96     return ret
97 # PIN validation
98 def pin( inVal, PIN ):
99     ret = 0; # no error 1, error 0
100    # while True:
101    if not re.match("[0-9]{4}$", inVal):
102        print ("Error! Make sure you only use numbers from 0–9 in PIN")
103        inVal = 'Z'
104        ret = 0
105    else:
106        if inVal == PIN :
107            #print("\nCorect PIN") # debug only
108            print("\n")
109            mainMenue()
110            inVal = 'Z'
111            ret = 1
112        # break
113    else:
114        print ("\nInvalid PIN!")
115        inVal = 'Z'
116        ret = 0
117    return ret

```

```

118
119 def delay( msec ):
120     cnt = 0
121     while cnt < msec:
122         cnt += 0.0001
123     return 1 # no error 1, error 0
124
125 def mainMenue( ):
126     print( " Main Menu" )
127     print( "_____ \n" )
128     print( " 1. CHECK BALANCE" )
129     print( " 2. WITHDRAW CASH" )
130     print( " 3. DEPOSIT CASH" )
131     print( " 4. CHANGE PIN" )
132     print( " 5. EXIT " )
133     return 1 # no error 1, error 0
134
135 def welcome( ):
136     print( "\n      $$$$$$      $$      $$$      $$      $$      $  " )
137     print( "      $      $      $$$$      $$ $      $$      $$      $  " )
138     print( "      $      $      $  $      $$ $      $$      $$      $  " )
139     print( "      $      $      $$  $$      $$      $      $$      $  " )
140     print( "      $      $      $$$  $$$      $$$      $$$      $$$  " )
141     print( "      $$$$$$      $$$  $$$      $$$      $$$      $$$  " )
142     print( "      $$$$$$      $$$$$$$$$$      $$$      $      $$$      " )
143     print( "      $      $      $$$      $$$      $$$      $      $$$      " )
144     print( "      $      $      $      $$$      $$$      $      $$$      " )
145     print( "      $      $      $      $$$      $$$      $      $$$      " )
146     print( "      $      $      $$$      $$$      $$$      $$$      $      " )
147     print( "      $$$$$$      $$$      $$$      $$$      $$$      $$$      \n" )
148     print( "      $$$      $$$      $$$$$$$$$$$$$$$$$$      " )
149     print( "      $$$ $      $$$      $$$$$$$$$$$$$$$$$$      " )
150     print( "      $$$ $      $ $$$      $$$      " )
151     print( "      $$$ $      $ $$$      $$$      " )
152     print( "      $$$ $      $ $$$      $$$      " )
153     print( "      $$$      $$$      $$$$$$$$$$$$$$$$$$      " )
154     print( "      $$$      $$$      $$$$$$$$$$$$$$$$$$      " )
155     print( "      $$$      $$$      $$$      " )
156     print( "      $$$      $$$      $$$      " )
157     print( "      $$$      $$$      $$$      " )
158     print( "      $$$      $$$      $$$$$$$$$$$$$$$$$$      " )
159     print( "      $$$      $$$      $$$$$$$$$$$$$$$$$$      " )
160     print( "_____")
161     print( "                Welcome to Bank Me  ")
162     print( "_____")
163     return 1 # no error 1, error 0

```

Listing 11: Python code for an ATM Part II functions.

### 4.3 Python code Part III

```

1  #-*- coding: utf-8 -*-
2  """
3  Spyder Editor
4
5  This is a temporary script file.
6  """
7  from time import asctime
8  import re
9  #from atm_class import welcome, pin, \
10 #selMainMenue, newPin, deposit, withdraw, balance, receipt, delay
11 from atm_class import Atm
12
13 atm = Atm(100,'2345') # instantiate class Atm to atm object
14
15 # Open a file
16 try:
17     fo = open("07_lab_error_log.txt", "a")

```

```

18     fr = open(atm.ReceiptFile, "w")
19     fo.write(asctime()+ ' ATM program starts '+ '\n')
20     fr.write('\n-----Bank Me-----\n')
21     fr.write('-----Transaction Receipt-----\n')
22     fr.write('*****\n\n')
23 except IOError as e:
24     print('File '+e.filename+' could not be opened or written to!')
25
26 # Star program
27 if atm.welcome(): # first method used from the atm object
28     error = 0
29 else:
30     error = 1
31     fo.write(asctime()+ ' welcome() error\n')
32
33 # PIN validation
34 while True:
35     ret = 0
36     if atm.pin( input("Please enter your PIN: "), atm.Pin):
37         error = 0
38         fo.write(asctime()+ ' User logged in \n')
39         break
40     else:
41         error = 1
42         fo.write(asctime()+ ' User login error\n')
43
44 # Main menu
45 while atm.Select != "5":
46     ret = 0
47     ret = atm.selMainMenu(input("PLEASE SELECT FROM THE ABOVE OTPIONS: ") )
48     # print(ret) # debug only
49     if (ret[0]):
50         error = 0
51         atm.Select = ret[1]
52     # print(ret[1] ) # debug only
53     else: # error
54         error = 1
55         atm.Select = 0
56         fo.write(asctime()+ ' User slection error\n')
57         print(ret[0])
58
59     if atm.Select == '5': # exit
60         while True:
61             if not re.match("[y]$", input('Do you wish to print the receipt , press y or n?
62             ')):
63                 print("\nThank you for chosing BANK ME!")
64                 break
65             else:
66                 atm.receipt(atm.ReceiptFile, fr)
67                 break
68         exit
69     elif atm.Select == '4': # change PIN
70         while True:
71             ret = 0
72             ret = atm.newPin(input("ENTER YOUR NEW PIN: "), fr)
73             # print(ret) # debug only
74             if (ret[0]):
75                 error = 0
76                 atm.Pin = ret[1]
77                 # print(ret[1] ) # debug only
78                 break
79             else: # error
80                 error = 1
81                 atm.Select = 0
82                 fo.write(asctime()+ ' User PIN error\n')
83                 break
84         elif atm.Select == '3': # Deposit

```



```

85         while True:
86             ret = 0
87             ret = atm.deposit(input("ENTER YOUR DEPOSIT AMOUNT IN $: ") \
88                               , atm.AMOUNT_MIN, atm.AMOUNT_MAX, atm.AccountValue, fr)
89 #         print(ret) # debug only
90         if (ret[0]):
91             error = 0
92             atm.AccountValue = ret[1]
93         #         print(ret[1] ) # debug only
94             break
95         else: # error
96             error = 1
97             atm.Select = 0
98             fo.write(asctime()+ ' Deposit error\n')
99
100     elif atm.Select == '2': # Withdraw
101         while True:
102             ret = 0
103             ret = atm.withdraw(input("ENTER YOUR WITHDRAW AMOUNT IN $: ") \
104                                , atm.AMOUNT_MIN, atm.AMOUNT_MAX, atm.AccountValue, fr)
105 #         print(ret) # debug only
106         if (ret[0]):
107             error = 0
108             atm.AccountValue = ret[1]
109         #         print(ret[1] ) # debug only
110             break
111         else: # error
112             error = 1
113             atm.Select = 0
114             fo.write(asctime()+ ' Withdraw error\n')
115
116     elif atm.Select == '1': # Balance
117         atm.balance(atm.AccountValue, fr)
118     else:
119         print("\nIncorrect Seleccion!")
120
121 fo.write(asctime()+ ' Program Closed\n')
122 fo.close() # to ensure file is closed
123 fr.close() # to ensure file is closed
124 atm.delay(3000)

```

Listing 12: Python instance for an ATM Part III.

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Oct 18 18:34:49 2018
4
5  @author: schwa
6  """
7  import re
8  class Atm:
9      AMOUNT_MIN = 0
10     AMOUNT_MAX = 1000
11
12     Pin = "1234"
13     Select = 0
14     AccountValue = 0
15     ReceiptFile = "atm_receipt.txt"
16
17     def __init__(self, AccountValue, Pin):
18         self.AccountValue = AccountValue
19         self.Pin = Pin
20
21     # Print Receipt
22     def receipt(self, filename, file):
23         file.write('THANK YOU FOR CHOOSING BANK ME! ')
24         file.close()
25         file = open(filename, "r")

```

```

26         for line in file:
27             print(line, end='')
28         file.close()
29         return 1
30     # Balance
31     def balance(self, account_value, file):
32         print("\nYOUR BALANCE IS $%.2f \n \n\r" % account_value)
33         file.write('Your checking account balance is————> $'\
34                     +str(account_value)+'\n\n')
35         self.delay(3000)
36         self.mainMenue( )
37         return 1
38     # Withdraw money
39     def withdraw(self, inVal, AMOUNT_MIN, AMOUNT_MAX, account_value, file ):
40         ret = [0, 0]; # no error 1, error 0
41         amount = 0
42         if not re.match("[0-9]*$", inVal):
43             amount = 0
44             print("\nInvalid Input only nmbers are alowed.\n")
45             ret=[0, inVal]
46         else:
47             amount = int(inVal)
48             if amount >= AMOUNT_MIN and amount <= account_value :
49                 account_value = account_value - amount
50                 print("\nWITHDRAW AMOUNT $%.2f\n" % amount)
51                 file.write('Your withdraw amounte is—————> $'\
52                             +str(amount)+'\n\n')
53                 amount = 0
54                 self.delay(3000)
55                 self.mainMenue( )
56                 ret=[1, account_value]
57             else:
58                 ret=[0, inVal]
59                 print("\nWITHDRAW AMOUNT $%.2f TOO BIG OR TOO SMALL\n" % amount)
60         return ret
61     # Deposit money
62     def deposit( self, inVal, AMOUNT_MIN, AMOUNT_MAX, account_value, file ):
63         ret = [0, 0]; # no error 1, error 0
64         amount = 0
65         if not re.match("[0-9]*$", inVal):
66             amount = 0
67             print("\nInvalid Input only numbers are alowed.\n")
68             ret=[0, inVal]
69         else:
70             amount = int(inVal)
71             if amount >= AMOUNT_MIN and amount <= AMOUNT_MAX :
72                 account_value = account_value + amount
73                 print("\nDEPOSIT AMOUNT $%.2f\n" % amount)
74                 file.write('Your deposit amounte is—————> $'\
75                             +str(amount)+'\n\n')
76                 amount = 0
77                 self.delay(3000)
78                 self.mainMenue( )
79                 ret=[1, account_value]
80             else:
81                 ret=[0, inVal]
82                 print("\nDEPOSIT AMOUNT $%.2f TOO BIG OR TOO SMALL\n" % amount)
83         return ret
84     # set new pin
85     def newPin( self, inVal, file ):
86         ret = [0, 0]; # no error 1, error 0
87         if not re.match("[0-9]{4}$", inVal):
88             print ("Error! Make sure you only use numbers from 0-9 in PIN")
89             self.delay(3000)
90             self.mainMenue( )
91             inVal = 'Z'
92             ret=[0, inVal]
93         else:

```



```

162     print("          $$ $          $$$  $$$$$$$$$$$$$$$$ ")
163     print("          $$ $      $ $$$  $$          ")
164     print("          $$ $      $ $$$  $$          ")
165     print("          $$      $ $ $$$  $$          ")
166     print("          $$      $$ $$$  $$$$$$$$$$$$$$$$ ")
167     print("          $$      $$$ $$$  $$$$$$$$$$$$$$$$ ")
168     print("          $$      $$$ $$$  $$          ")
169     print("          $$      $$$ $$$  $$          ")
170     print("          $$      $$$ $$$  $$          ")
171     print("          $$      $$$ $$$$$$$$$$$$$$$$ ")
172     print("          $$      $$$ $$$$$$$$$$$$$$$$ ")
173     print("=====")
174     print("          Welcome to Bank Me ")
175     print("=====")
176     return 1 # no error 1, error 0

```

Listing 13: Python class for an ATM Part III.