|  |
| --- |
|  |

|  |
| --- |
| **QA Consulting.** |
| Files |
| Exercise Guide 5 |

# Exercise 5.1 – Write Teams

Create:

* New folder and move to: **05Files\Exercise**
* New file: **01WriteTeams.py**

Code a program that:

* Opens a file called **teams.txt**
* Inputs and writes the names of 5 teams to the file

Save and run:

* Check the contents of the file using Notepad

# Exercise 5.2 – Read Teams

Create:

* New file: **02ReadTeams.py**

Code a program that:

* Opens a file called teams.txt
* Reads and outputs the names of the first 2 teams in the file
* Reads and outputs the names of the rest of the teams in the file

Save and run.

# Exercise 5.3 – Read By Line Teams

Create:

* New file: **03ReadByLineTeams.py**

Code a program that:

* Opens a file called teams.txt
* Reads and outputs the names of the 3rd team in the file twice

Save and run.

# Exercise 5.4 – Read By Character Teams

Create:

* New file: **04ReadByCharacterTeams.py**

Code a program that:

* Opens a file called teams.txt
* Reads and outputs the second and the third characters of the 3rd team in the file

Save and run.

# Exercise 5.5 – Edit Teams

Create:

* New file: **05EditTeams.py**

Code a program that:

* Opens a file called teams.txt
* Edits the file to include just the second and fourth teams

Save and run:

* Check the contents of the file using Notepad

# Exercise 5.6 – ATM Advanced

Create

* New file: **06ATMAdvanced.py**

Code a program that:

* Simulates the behaviour of an Automatic Teller Machine (ATM) or Cash Machine
* See full specification over page

Save and run:

* Check the contents of the file using Notepad

Northern Frock needs you to write a program for their new ATMs (or Automatic Teller Machines). In this version, you will need to use a customer database that will initially look like this (ignore the top row):

ID Title First Name Surname Balance

1057 Mr. Jeremy Clarkson 172.16

2736 Miss Suzanne Perry 15.62

4659 Mrs. Vicki Butler-Henderson 23.91

5691 Mr. Jason Plato 62.17

Write a program that will do the following:

* Generate the above data file.
* Prompt the user for their ID number, validating against the list of known users.
* If the ID number 99999 (5x 9s) is given the machine should shut down.
* Print a menu:

Welcome to Northern Frock

* 1 - Display balance
* 2 - Withdraw funds
* 3 - Deposit funds
* 9 - Return card
* Enter an option:
* If ‘1’ is entered, display the current balance and the maximum amount available for withdrawal (must be a multiple of £10)
* If ‘2’ is entered, provide a sub-menu with withdrawal amounts:
* Please select withdrawal amount
* 1 - £10
* 2 - £20
* 3 - £40
* 4 - £60
* 5 - £80
* 6 - £100
* 7 - Other amount
* 8 - Return to main menu
* Enter an option:
  + If ‘1 to 6’’ is selected check that the requested withdrawal is allowed, print a message to show that the money has been withdrawn, calculate the new balance and return to main menu.
  + If ‘7’ is selected, then prompt the user for an integer value. Check this number is a multiple of 10 and that the withdrawal is permitted, print a message to show that the money has been withdrawn, calculate the new balance and return to main menu.
  + If ‘8’ is selected return to main menu.
* If ‘3’ is entered, provide another menu that will allow the user to enter an amount to deposit (does not need to be a multiple of £10), return to main menu or return card. If funds are deposited, provide appropriate feedback and update the balance.
* If ‘9’ is entered, print a goodbye message and return to the initial prompt (for the next customer).
* If another value is entered, print an error message and print the menu again.

Extensions:

* Add an extra field to the database for a 4-digit PIN which should be prompted for and checked following the entry of the ID number. The user should also have the option to change their PIN.
* Add another field to record whether the card is blocked. Three incorrect PIN attempts should permanently lock the card. PIN attempts should only be reset by correctly entering the PIN. Simply removing the card and trying again should not work.
* Create an encoding algorithm that will store the PIN in a format that cannot be read by just looking at the file. Only by running the PIN attempt through the same algorithm can the match be found.

# Solutions

## Solution 5.1 – Write Teams

# Program : 01WriteTeams

# Author : John Merchant

# Date : 12 Jul 2016

# Purpose : Exercise to open and write to file

file = open("teams.txt","w")

for i in range(5):

team = input("Please input team: ") + "\n"

file.write(team)

file.close()

## Solution 5.2 – Read Teams

# Program : 02ReadTeams

# Author : John Merchant

# Date : 12 Jul 2016

# Purpose : Exercise to open and read file

file = open("teams.txt","r")

print("First line:")

print(file.readline())

print("Second line:")

print(file.readline())

print("Rest of file:")

print(file.read())

file.close()

## Solution 5.3 – Read By Line Teams

# Program : 03ReadByLineTeams

# Author : John Merchant

# Date : 12 Jul 2016

# Purpose : Exercise to open and read file by line

file = open("teams.txt","r")

file.readline()

file.readline()

print("Third line:")

print(file.readline())

file.seek(0)

file.readline()

file.readline()

print("Third line again")

print(file.readline())

file.close()

## Solution 5.4 – Read By Character Teams

# Program : 04ReadByCharacterTeams

# Author : John Merchant

# Date : 12 Jul 2016

# Purpose : Exercise to open and read file by character

file = open("teams.txt","r")

file.readline()

file.readline()

file.read(1)

print("2nd and 3rd Characters of 3rd team:")

print(file.read(2))

file.close()

## Solution 5.5 – Edit Teams

# Program : 05EditTeams

# Author : John Merchant

# Date : 12 Jul 2016

# Purpose : Exercise to open and edit file

file = open("teams.txt","r")

outfile = ""

file.readline()

outfile = outfile + file.readline()

file.readline()

outfile = outfile + file.readline()

file = open("teams.txt","w")

file.write(outfile)

file.close()