**COMPUTER SCIENCE**

**PROJECT FILE**

**ON**

**ATM-SYSYTEM**

****

**PROJECT PREPARED BY:**

1. **B THARUN**

**XII A**

**Session: 2019-20120**

**K V SOUTHERN COMMAND**

**School**

**TABLE OF CONTENTS**

* **Certificate**
* **Acknowledgement**
* **Overview**
* **Coding**
* **Limitations**
* **Output**
* **Requirements**
* **Bibliography**

**Certificate**

**This is to certify that L.B THARUN of class twelve, K V SOUTHERN COMMAND School, Pune has successfully completed his project in computer practicals for the CBSE board as prescribed by CBSE in the year 20019-2020.**

**Date :18-01-2020**

**Roll No. :15617069**

**Signature of Internal Signature of External Examiner Examiner**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Acknowledgement**

**I thank my Computer Science teacher Mr.Rajkamal Alaria for guidance and support. I also thank my Principal Mrs. Snehal Marathe. I would also like to thank my parents and my friend for encouraging me during the course of this project. Finally I would like to thank CBSE for giving me this opportunity to undertake this project.**

**Overview**

# Python

**Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language" \o "Interpreted language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language" \o "High-level programming language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language" \o "General-purpose programming language) [programming language](https://en.wikipedia.org/wiki/Programming_language" \o "Programming language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum" \o "Guido van Rossum) and first released in 1991, Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability" \o "Code readability) with its notable use of [significant whitespace](https://en.wikipedia.org/wiki/Off-side_rule" \o "Off-side rule). Its language constructs and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming" \o "Object-oriented programming) approach aim to help programmers write clear, logical code for small and large-scale projects.[[28]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-AutoNT-7-28)

Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language" \o "Dynamic programming language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)" \o "Garbage collection (computer science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms" \o "Programming paradigms), including [procedural](https://en.wikipedia.org/wiki/Procedural_programming" \o "Procedural programming), object-oriented, and [functional programming](https://en.wikipedia.org/wiki/Functional_programming" \o "Functional programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library" \o "Standard library).[[29]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-About-29)

Python was conceived in the late 1980s as a successor to the [ABC language](https://en.wikipedia.org/wiki/ABC_(programming_language)" \o "ABC (programming language)). Python 2.0, released in 2000, introduced features like [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension" \o "List comprehension) and a [garbage collection](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)" \o "Garbage collection (computer science)) system capable of collecting [reference cycles](https://en.wikipedia.org/wiki/Reference_cycle" \o "Reference cycle). Python 3.0, released in 2008, was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility" \o "Backward compatibility), and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on January 1, 2020 (first planned for 2015) after which security patches and other improvements will not be released for it.[[30]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-30)[[31]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-31) With Python 2's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)" \o "End-of-life (product)), only Python 3.5.x[[32]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-32) and later are supported.

Python [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)" \o "Interpreter (computing)) are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system" \o "Operating system). A global community of programmers develops and maintains [CPython](https://en.wikipedia.org/wiki/CPython" \o "CPython), an [open source](https://en.wikipedia.org/wiki/Open-source_software" \o "Open-source software)[[33]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-33) [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation" \o "Reference implementation). A [non-profit organization](https://en.wikipedia.org/wiki/Nonprofit_organization" \o "Nonprofit organization), the [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation" \o "Python Software Foundation), manages and directs resources for Python and CPython development

CODING

import getpass

import string

import os

# creatinga lists of users, their PINs and bank statements

users = ['jaya', 'ruchi', 'anu']

pins = ['1111', '2222', '3333']

amounts = [1000, 2000, 3000]

count = 0

#welcomeing the user to the atm

print('-----------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('---WELCOME TO THE TCS BANK"s ATM---')

print("-----\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--------")

print('----\*\*\*\*\*\*\*\*THARUN\*\*\*\*\*\*\*\*---------')

print("-----\*\*\*\*\*CHAMPSTAR\*\*\*\*\*\*\*---------")

print('-----\*\*\*\*\*\*\*\*BANK\*\*\*\*\*\*\*\*\*\*--------')

print("-----\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-----")

print('!!!PLEASE ENTER YOUR USER BELLOW!!!')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('-----------------------------------')

# while loop checks existance of the enterd username

while True:

user = input('\nENTER USER NAME: ')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print("---WELCOME TO THE TCS's BANKING SYSTEM---")

user = user.lower()

if user in users:

if user == users[0]:

n = 0

elif user == users[1]:

n = 1

else:

n = 2

break

else:

print('----------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('INVALID USERNAME')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('----------------')

# comparing pin

while count < 3:

print('------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

pin = str(input('PLEASE ENTER PIN: '))

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('------------------')

if pin.isdigit():

if user == users[0]:

if pin == pins[0]:

break

else:

count += 1

print('-----------')

print('\*\*\*\*\*\*\*\*\*\*\*')

print('INVALID PIN')

print('\*\*\*\*\*\*\*\*\*\*\*')

print('-----------')

print()

if user == users[1]:

if pin == pins[1]:

break

else:

count += 1

print('-----------')

print('\*\*\*\*\*\*\*\*\*\*\*')

print('INVALID PIN')

print('\*\*\*\*\*\*\*\*\*\*\*')

print('-----------')

print()

if user == users[2]:

if pin == pins[2]:

break

else:

count += 1

print('-----------')

print('\*\*\*\*\*\*\*\*\*\*\*')

print('INVALID PIN')

print('\*\*\*\*\*\*\*\*\*\*\*')

print('-----------')

print()

else:

print('------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('PIN CONSISTS OF 4 DIGITS')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('------------------------')

count += 1

# in case of a valid pin- continuing, or exiting

if count == 3:

print('-----------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('3 UNSUCCESFUL PIN ATTEMPTS, EXITING')

print('!!!!!YOUR CARD HAS BEEN LOCKED!!!!!')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('-----------------------------------')

exit()

print('-------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('LOGIN SUCCESFUL, CONTINUE')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('-------------------------')

print()

print('--------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(str.capitalize(users[n]), 'welcome to TCS ATM')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('--------TCS ATM SYSTEM-----------')

# Main menu

while True:

#os.system('clear')

print('-------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

response = input('SELECT FROM FOLLOWING OPTIONS: \nStatement\_\_(S) \nWithdraw\_\_\_(W) \nLodgement\_\_(L) \nChange PIN\_(P) \nQuit\_\_\_\_\_\_\_(Q) \n: ').lower()

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('-------------------------------')

valid\_responses = ['s', 'w', 'l', 'p', 'q']

response = response.lower()

if response == 's':

print('---------------------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(str.capitalize(users[n]), 'YOU HAVE ', amounts[n],'RUPEES ON YOUR ACCOUNT.')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('---------------------------------------------')

elif response == 'w':

print('---------------------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

cash\_out = int(input('ENTER AMOUNT YOU WOULD LIKE TO WITHDRAW: '))

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('---------------------------------------------')

if cash\_out%10 != 0:

print('------------------------------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*')

print('AMOUNT YOU WANT TO WITHDRAW MUST TO MATCH 10 RUPEES NOTES')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*')

print('------------------------------------------------------')

elif cash\_out > amounts[n]:

print('-----------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('YOU HAVE INSUFFICIENT BALANCE')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('-----------------------------')

else:

amounts[n] = amounts[n] - cash\_out

print('-----------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('YOUR NEW BALANCE IS: ', amounts[n], 'RUPEES')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('-----------------------------------')

elif response == 'l':

print()

print('---------------------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

cash\_in = int(input('ENTER AMOUNT YOU WANT TO LODGE: '))

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('---------------------------------------------')

print()

if cash\_in%10 != 0:

print('----------------------------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*')

print('AMOUNT YOU WANT TO LODGE MUST TO MATCH 10 RUPEES NOTES')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*')

print('----------------------------------------------------')

else:

amounts[n] = amounts[n] + cash\_in

print('----------------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('YOUR NEW BALANCE IS: ', amounts[n], 'RUPEES')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('----------------------------------------')

elif response == 'p':

print('-----------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

new\_pin = str(input('ENTER A NEW PIN: '))

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('-----------------------------')

if new\_pin.isdigit() and new\_pin != pins[n] and len(new\_pin) == 4:

print('------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

new\_ppin = str(input('CONFIRM NEW PIN: '))

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('-------------------')

if new\_ppin != new\_pin:

print('------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*')

print('PIN MISMATCH')

print('\*\*\*\*\*\*\*\*\*\*\*\*')

print('------------')

else:

pins[n] = new\_pin

print('NEW PIN SAVED')

else:

print('-------------------------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(' NEW PIN MUST CONSIST OF 4 DIGITS \nAND MUST BE DIFFERENT TO PREVIOUS PIN')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('-------------------------------------')

elif response == 'q':

exit()

else:

print('------------------')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('RESPONSE NOT VALID')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('------------------')

**Limitations**

**---------------------------------------------**

**---------------------------------------------**

**-------Welcome To TCS's Banking system-------**

**-----\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-----**

**-----\*\*\*\*\*\*\*\*\*\*\*\*\*THARUN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-----**

**-----\*\*\*\*\*\*\*\*\*\*\*\*CHAMPSTAR\*\*\*\*\*\*\*\*\*\*\*\*\*\*-----**

**-----\*\*\*\*\*\*\*\*\*\*\*\*\*\*BANK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-----**

**-----\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-----**

**---------------------------------------------**

**---------------------------------------------**

**---------------------------------------------**

**---------------------------------------------**

**Following people are your esteemed users**

|  |  |
| --- | --- |
| **USERNAME** | **PIN** |
| **JAYA** | **1111** |
| **RUCHI** | **2222** |
| **ANU** | **3333** |

**opening balance are(1000,2000,3000) respectively**

**conditions**

**{--------SELECT FROM FOLLOWING OPTIONS:**

**Statement\_\_(S)**

**Withdraw\_\_\_(W)**

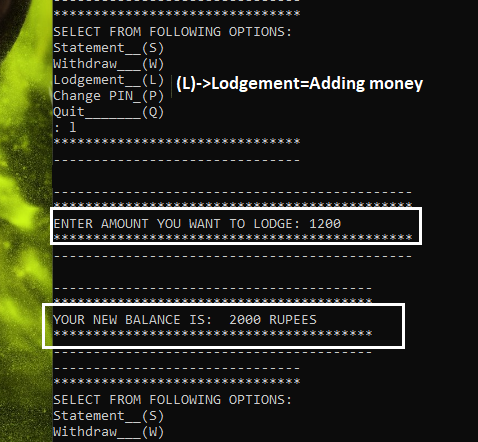
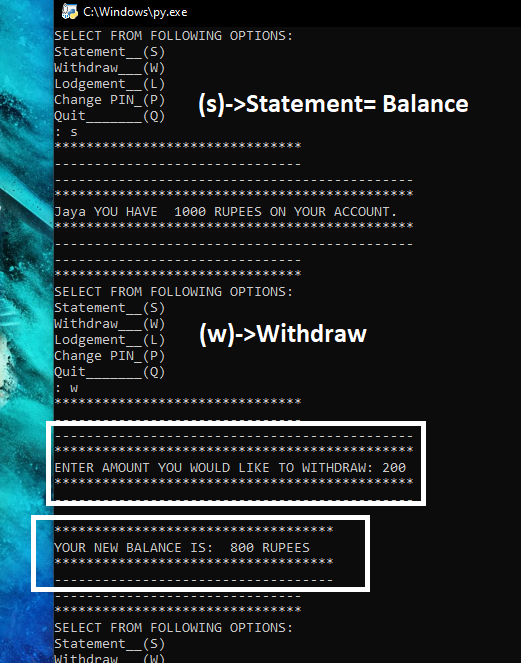
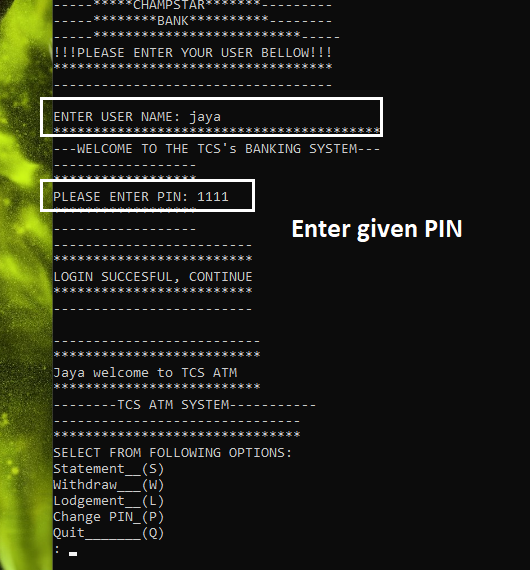
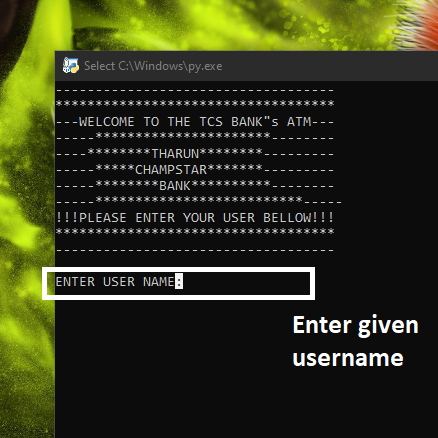
**Lodgement\_\_(L)**

**Change PIN\_(P)**

**Quit\_\_\_\_\_\_\_(Q)**

**------------}**

**use only given letters(s,w,l,p,q)**

**Output**

**Requirements**

* **Hardware Required**
* **Printer, to print the required documents of the project**
* **Compact Drive**
* **Processor : Pentium III**
* **Ram : 64 MB**
* **Hard disk : 20 GB.**
* **software Required**
* **Operating system : Windows XP(at-least)**
* **Python 2.7,(min) for execution of program**
* **Ms word, for presentation of output.**

**BIBLIOGRAPHY**

**j0292020**

**COMPUTER SCIENCE IN PYTHON BY :– SUMITA ARORA**

**en.wikipedia.org/wiki/Python\_(programming\_language)**