VIT Vidyalankar Institute of Technology ACCREDITED A+ BY NAAC	**Department of Information Technology**

Semester	S.E. Semester IV – INFT
Subject	Python Lab
Laboratory Teacher:	Prof. Shruti Agrawal
Laboratory	-LAB07

Student Name	Jayant Dethe	
Roll Number	22101A0065	
Grade and Subject Teacher's Signature		

Experiment Number	11	
Problem Statement	Write a python program to perform exception handling using try, except, finally, else, assert and raise.	
Resources / Apparatus Required	Hardware: Desktop/Laptop Laptop	Software: REPLIT

Code:	# Function to divide two numbers
	def divide(a, b):
	try:
	result = a / b
	except ZeroDivisionError:
	print("Error: Division by zero!")
	else:	

```
        print("Result of division:", result)
    finally:
        print("Division operation completed.")


# Function to check if a number is positive using assert
def check_positive(num):
    try:
        assert num > 0, "Number must be positive"
    except AssertionError as e:
        print("AssertionError:", e)
    else:
        print("Number is positive:", num)


# Function to demonstrate custom exception using raise
def withdraw(total_balance, withdrawal_amount):
    try:
        if withdrawal_amount <= 0:
            raise ValueError("Withdrawal amount must be greater than
zero")
        if withdrawal_amount > total_balance:
            raise ValueError("Insufficient balance")
        print("Withdrawal successful. Remaining balance:",
total_balance - withdrawal_amount)
    except ValueError as e:
        print("ValueError:", e)


# Main function
def main():
```

```
    # Testing divide function
    divide(20, 4)
    divide(20, 0)

    # Testing assert function
    check_positive(7)
    check_positive(-9)

    # Testing raise function
    withdraw(2000, 800)
    withdraw(2000, -300)
    withdraw(2000, 2500)

if __name__ == "__main__":
    main()
```

Output:

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

=== RESTART: C:/Users/dethe/AppData/Local/Programs/Python/Python312/jd_13.py ===
Result of division: 5.0
Division operation completed.
Error: Division by zero!
Division operation completed.
Number is positive: 7
AssertionError: Number must be positive
Withdrawal successful. Remaining balance: 1200
ValueError: Withdrawal amount must be greater than zero
ValueError: Insufficient balance
```

VIT Vidyalankar Institute of Technology ACCREDITED A+ BY NAAC	**Department of Information Technology**

Semester	S.E. Semester IV – INFT
Subject	Python Lab
Laboratory Teacher:	Prof. Shruti Agrawal
Laboratory	-LAB07

Student Name	Jayant Dethe	
Roll Number	22101A0065	
Grade and Subject Teacher's Signature		

Experiment Number	12	
Problem Statement	Write a python program to perform basic operations on file and also perform pickle and unpickle operations in it.	
Resources / Apparatus Required	Hardware: Desktop/Laptop Laptop	Software: REPLIT

Code:	import pickle
	
	# Function to write data to a file
	def write_data(file_name, content):
	  with open(file_name, 'wb') as file:
	  pickle.dump(content, file)
	  print(f"Data written to {file_name} successfully.")	

```
# Function to read data from a file
def read_data(file_name):
    with open(file_name, 'rb') as file:
        content = pickle.load(file)
    return content

# Function to append data to a file
def append_data(file_name, additional_data):
    with open(file_name, 'ab') as file:
        pickle.dump(additional_data, file)
    print(f"Data appended to {file_name} successfully.")

# Main function to demonstrate file operations
def main():
    # File name
    file_name = "data.pkl"

    # Data to be written
    data_to_write = ["carrot", "potato", "tomato"]

    # Writing data to the file
    write_data(file_name, data_to_write)

    # Reading data from the file
    data_read = read_data(file_name)
    print("Data read from file:", data_read)
```

```
    # Appending more data to the file
    more_data_to_append = ["lettuce", "cucumber"]
    append_data(file_name, more_data_to_append)

    # Reading data from the file after appending
    data_read_after_append = read_data(file_name)
    print("Data read from file after appending:",
data_read_after_append)

if __name__ == "__main__":
    main()
```

Output:

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:/Users/dethe/AppData/Local/Programs/Python/Python312/jd_13.py
Data written to data.pkl successfully.
Data read from file: ['carrot', 'potato', 'tomato']
Data appended to data.pkl successfully.
Data read from file after appending: ['carrot', 'potato', 'tomato']
```

VIT Vidyalankar Institute of Technology ACCREDITED A+ BY NAAC	**Department of Information Technology**

Semester	S.E. Semester IV – INFT
Subject	Python Lab
Laboratory Teacher:	Prof. Shruti Agrawal
Laboratory	-LAB07

Student Name	Jayant Dethe	
Roll Number	22101A0065	
Grade and Subject Teacher's Signature		

Experiment Number	13	
Problem Statement	Write a python program to create a regular expression:- 1- To extract only name but not number from a given string. 2- To retrive date of birth from the string 3- To retrive all the words starting with "a" in a given string.	
Resources / Apparatus Required	Hardware: Desktop/Laptop Laptop	Software: PyCharm
Code:	import re # 1. Regular expression to extract only name but not number from a given string:- def extract_name(input_text):	

	name_pattern = r'\b[A-Za-z]+\b'
	  # Matches one or more alphabetical characters
	  names = re.findall(name_pattern, input_text)
	  return names
	
	# 2. Regular expression to retrieve date of birth from the string:-
	def retrieve_dob(input_text):
	  dob_pattern = r'\d{2}-\d{2}-\d{4}'
	  # Matches date in dd-mm-yyyy format
	  dob = re.findall(dob_pattern, input_text)
	  return dob
	
	# 3. Regular expression to retrieve all the words starting with "a" in a given string:-
	def retrieve_words_starting_with_a(input_text):
	  words_starting_with_a_pattern = r'\b[aA]\w+'
	  # Matches words starting with 'a' or 'A'
	  words = re.findall(words_starting_with_a_pattern, input_text)
	  return words
	
	# Example usage
	sample_text = "Jayant was born on 26-10-2004. His friend Harshal likes to play basketball"
	print("Names:", extract_name(sample_text))
	print("Date of Birth:", retrieve_dob(sample_text))
	print("Words starting with 'a':", retrieve_words_starting_with_a(sample_text))

Output:	Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information. === RESTART: C:\Users\dethe\AppData\Local\Programs\Python\Python312\jd_13.py === Names: ['Jayant', 'was', 'born', 'on', 'His', 'friend', 'Harshal', 'likes', 'to', 'play', 'basketball'] Date of Birth: ['26-10-2004'] Words starting with 'a': []