|  |  |  |
| --- | --- | --- |
| **Sinhgad Technical Education Society’s Sinhgad Institute of Business Administration and Research Kondhwa (Bk.), Pune 411048 Subject : IT-21 Practicals (Python Programming)** | | |
| **Sr. No** | **Assignment Questions** | **Sign** |
| **1** | Write a program to check whether the given number is prime or not? |  |
| **2** | Write a program to print all perfect nos given range? |  |
| **3** | Write a program to print sum of digit of any number |  |
| **4** | Write a program to create tuple of name of 5 fruits and print them using two ways  1. Directly print all the item  2. Print one by one item in the tuple |  |
| **5** | Write a Python function to multiply all the numbers in a list.  Sample List : [8, 2, 3, -1, 7]Expected Output : -336 |  |
| **6** | Write a Python program to reverse a string.  Sample String : "1234abcd"  Expected Output : "dcba4321" |  |
| **7** | Write a program to find out sum of 2 numbers 3 numbers 4 numbers and 5 numbers using default parameter passing. |  |
| **8** | Write a program to find out multiplication of 2 numbers 3 numbers 4 numbers and 5 numbers using default parameter passing. |  |
| **9** | Write a program to find out area circumference of circle using lambda function |  |
| **10** | Write a program to pass variable length arguments (dictionary and touple to print email |  |
| **11** | Write a program to create class dog to store Breed Age and Color attribute and display it. |  |
| **12** | Write a program to create class person to store name,age,sex of the person use constructors to initialize objects |  |
| **13** | Write a program to find following pattern from the string “Sky is only Limit of the Programmer in Year 2023”  1.      All capital alphabets  2.      All digits  3.      All small alphabets  4.      All lower case between “m-“ to “s” |  |
| **14** | Write a program to handle name error exception |  |
| **15** | Write a program to read any text file and print report of total characters as follows  Character Types  Capital letters  Small letters  Digits  Special Characters |  |
| **16** | Write a program to read any text file and convert all small case letters to capital letters |  |
| **17** | Write a program to read any text file and convert all capital case letters to small letters |  |
| 18 | Write a Python program to create "employee" collection with fields (ID, name, address, phone, email, dept) in MongoDB. Perform the following operations-  i. Insert minimum 5 documents in employee collection.  ii. Display all employees in "Accounts" department.  iii. Delete employee with ID - 210345.  iv. Update phone with new phone for employee Id- 123 |  |
| 19 | Write a Python program to create "movie" collection with fields (movie\_id, movie\_name, movie\_type, movie\_year,  i. Insert minimum 5 documents in movie collection.  ii. Display the list of all movies released in year 2020.  iii. Update the movie \_type to "Comedy" wherever movie\_lead\_actor is "Govinda".  iv. Display the list of movies in descending order of movie\_year.  v. Delete movie having movie\_director as "David Dhavan". |  |
| 20 | Write a Python program to create "Library" database. In Library database create 2 collections such as "Book" with fields (book\_id, title, author, publisher, price, copies) and "Transaction" with fields (trans\_id, trans\_type, trans\_date, stud\_id, book\_id). Perform the following operations -  i. Insert minimum 5 documents in Book & Transaction collection.  ii. Display the list of books in Library in ascedning order of Title.  iii. Display the trasaction details for the transactions done on "18-06-2023".  iv. Update the copies of books to 10 for the subject of Java.  v. Delete the trasactions done in year 2020. |  |
| 21 | Write a menu driven(add subtract, multiply and division, exit) program to handle two 1 D array |  |

1.Write a program to check whether the given number is prime or not?

Code:

def check\_even\_odd(number):

if number % 2 == 0:

print("The number is even.")

else:

print("The number is odd.")

# Prompt the user to enter a number

number = int(input("Enter a number: "))

# Check and display the result

check\_even\_odd(number)

1. Write a program to print all perfect nos given range?

Ans:

n = int(input("Enter any number: "))

sum1 = 0

**for** i **in** range(1, n):

**if**(n % i == 0):

sum1 = sum1 + i

**if** (sum1 == n):

**print**("The number is a Perfect number!")

**else**:

**print**("The number is not a Perfect number!")

1. Write a program to print sum of digit of any number

Ans

num = input("Enter Number: ")

sum = 0

for i in num:

sum = sum + int(i)

print(sum)

1. Write a program to create tuple of name of 5 fruits and print them using two ways

1. Directly print all the item

2. Print one by one item in the tuple

Ans:

fruits = ('apple', 'banana', 'orange', 'kiwi', 'grape')

# Directly print all the items

print(fruits)

# Print one by one item in the tuple

for fruit in fruits:

print(fruit)

1. Write a Python function to multiply all the numbers in a list.   
   *Sample List* : [8, 2, 3, -1, 7]*Expected Output* : -336

def multiply\_list(numbers):

result = 1

for num in numbers:

result \*= num

return result

numbers = [8, 2, 3, -1, 7]

product = multiply\_list(numbers)

print("Product of", numbers, "is", product)

1. Write a Python program to reverse a string.   
   *Sample String* : "1234abcd"  
   *Expected Output* : "dcba4321"

Ans: string = input("Enter a string: ")

reverse\_string = string[::-1]

print("Original string:", string)

print("Reversed string:", reverse\_string)

1. Write a program to find out sum of 2 numbers 3 numbers 4 numbers and 5 numbers using default parameter passing.

Ans: def sum\_numbers(a, b, c=0, d=0, e=0):

return a + b + c + d + e

# Sum of 2 numbers

print("Sum of 2 numbers:", sum\_numbers(10, 20))

# Sum of 3 numbers

print("Sum of 3 numbers:", sum\_numbers(10, 20, 30))

# Sum of 4 numbers

print("Sum of 4 numbers:", sum\_numbers(10, 20, 30, 40))

# Sum of 5 numbers

print("Sum of 5 numbers:", sum\_numbers(10, 20, 30, 40, 50))

1. Write a program to find out multiplication of 2 numbers 3 numbers 4 numbers and 5 numbers using default parameter passing.

Ans: def multiply\_numbers(a, b, c=1, d=1, e=1):

return a \* b \* c \* d \* e

# Multiplication of 2 numbers

print("Multiplication of 2 numbers:", multiply\_numbers(10, 20))

# Multiplication of 3 numbers

print("Multiplication of 3 numbers:", multiply\_numbers(10, 20, 30))

# Multiplication of 4 numbers

print("Multiplication of 4 numbers:", multiply\_numbers(10, 20, 30, 40))

# Multiplication of 5 numbers

print("Multiplication of 5 numbers:", multiply\_numbers(10, 20, 30, 40, 50))

1. Write a program to find out area circumference of circle using lambda function

Ans:

radius = float(input("Enter the radius of the circle: "))

# Lambda function to calculate area of circle

area = lambda r: 3.14159 \* r \*\* 2

# Lambda function to calculate circumference of circle

circumference = lambda r: 2 \* 3.14159 \* r

print("Area of the circle:", area(radius))

print("Circumference of the circle:", circumference(radius))

1. Write a program to pass variable length arguments (dictionary and touple to print email

Ans: def print\_email(\*args):

for arg in args:

if isinstance(arg, dict):

# If argument is a dictionary

print(arg['name'] + "'s email is " + arg['email'])

elif isinstance(arg, tuple):

# If argument is a tuple

print(arg[0] + "'s email is " + arg[1])

# Example usage

person1 = {'name': 'John', 'email': 'john@example.com'}

person2 = ('Alice', 'alice@example.com')

print\_email(person1, person2)

1. Write a program to create class dog to store Breed Age and Color attribute and display it.

Ans: class Dog:

def \_\_init\_\_(self, breed, age, color):

self.breed = breed

self.age = age

self.color = color

def display\_info(self):

print("Breed:", self.breed)

print("Age:", self.age)

print("Color:", self.color)

# Example usage

my\_dog = Dog("Labrador Retriever", 4, "Yellow")

my\_dog.display\_info()

1. Write a program to create class person to store name,age,sex of the person use constructors to initialize objects

Ans: class Person:

def \_\_init\_\_(self, name, age, sex):

self.name = name

self.age = age

self.sex = sex

def display\_info(self):

print(f"Name: {self.name}")

print(f"Age: {self.age}")

print(f"Sex: {self.sex}")

# Example usage:

person1 = Person("John", 30, "Male")

person2 = Person("Jane", 25, "Female")

person1.display\_info()

person2.display\_info()

1. Q1. Write a program to find following pattern from the string “Sky is only Limit of the Programmer in Year 2023”
2. All capital alphabets
3. All digits
4. All small alphabets
5. All lower case between “m-“ to “s”

Ans

1. To find all capital alphabets:

string = "Sky is only Limit of the Programmer in Year 2023"

capital\_letters = ""

for char in string:

if char.isupper():

capital\_letters += char

print(capital\_letters)

Output: S L P Y Y

2. To find all digits:

string = "Sky is only Limit of the Programmer in Year 2023"

digits = ""

for char in string:

if char.isdigit():

digits += char

print(digits)

Output: 2023

3. To find all small alphabets:

string = "Sky is only Limit of the Programmer in Year 2023"

small\_letters = ""

for char in string:

if char.islower():

small\_letters += char

print(small\_letters)

Output: k y i s o n l y m i t o f t h e r o g r a m m e r i n

4. To find all lower case between “m-“ to “s”:

string = "Sky is only Limit of the Programmer in Year 2023"

lowercase\_m\_to\_s = ""

start\_index = string.index("m")

end\_index = string.index("s")

for i in range(start\_index, end\_index):

if string[i].islower():

lowercase\_m\_to\_s += string[i]

print(lowercase\_m\_to\_s)

Output: myionlyitoftheogramein

1. Write a program to handle name error exception

Ans

try:

print(variable\_name) # variable\_name is not defined

except NameError:

print("Variable is not defined")

# Output: Variable is not defined

1. Write a program to read any text file and print report of total characters as follows

File report of a.txt

Character Types Total Count

Capital letters 22

Small letters 123

Digits 34

Special Characters 20

Ans

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

# Write content to the file

file.write("This is some example text 123")

# Close the file

file.close()

>> def count\_characters(filename):

character\_types = {

"Capital letters": 0,

"Small letters": 0,

"Digits": 0,

"Special Characters": 0

}

with open(filename, "r") as file:

data = file.read()

for char in data:

if char.isupper():

character\_types["Capital letters"] += 1

elif char.islower():

character\_types["Small letters"] += 1

elif char.isdigit():

character\_types["Digits"] += 1

else:

character\_types["Special Characters"] += 1

return character\_types

# Example usage:

filename = "a.txt" # Replace with the path to your file

character\_count = count\_characters(filename)

print(f"File report of {filename}")

print("Character Types\t\t\tTotal Count")

print("Capital letters\t\t\t", character\_count["Capital letters"])

print("Small letters\t\t\t", character\_count["Small letters"])

print("Digits\t\t\t\t\t", character\_count["Digits"])

print("Special Characters\t\t", character\_count["Special Characters"])

1. Write a program to read any text file and convert all small case letters to capital letters

Ans:

try:

file = open("example.txt", "r") # replace example.txt with your file name

text = file.read()

file.close()

text = text.upper() # convert all small case letters to capital letters

print(text)

except FileNotFoundError:

print("File not found")

except:

print("An error occurred")

1. Write a program to read any text file and convert all capital case letters to small letters

Ans:

try:

file = open("example.txt", "r") # replace example.txt with your file name

text = file.read()

file.close()

text = text.lower() # convert all capital letters to small letters

print(text)

except FileNotFoundError:

print("File not found")

except:

print("An error occurred")

1. Write a Python program to create "employee" collection with fields (ID, name, address, phone, email, dept) in MongoDB. Perform the following operations-

i. Insert minimum 5 documents in employee collection.

ii. Display all employees in "Accounts" department.

iii. Delete employee with ID - 210345.

iv. Update phone with new phone for employee Id- 123

ans

pip install pymongo

import pymongo

client=pymongo.MongoClient("mongodb+srv://sgausiya28:root@cluster0.czdcv6d.mongodb.net/?retryWrites=true&w=majority")

db=client['db1']

employee\_collection=db['emp']

# i. Insert minimum 5 documents in employee collection

employees = [

{

"ID": 123,

"name": "John Doe",

"address": "123 Main St, City",

"phone": "9876543210",

"email": "john.doe@example.com",

"dept": "Accounts"

},

{

"ID": 124,

"name": "Jane Smith",

"address": "456 Elm St, City",

"phone": "9876543211",

"email": "jane.smith@example.com",

"dept": "Accounts"

},

{

"ID": 125,

"name": "Tom Johnson",

"address": "789 Oak St, City",

"phone": "9876543212",

"email": "tom.johnson@example.com",

"dept": "Sales"

},

{

"ID": 126,

"name": "Emily Davis",

"address": "321 Pine St, City",

"phone": "9876543213",

"email": "emily.davis@example.com",

"dept": "HR"

},

{

"ID": 127,

"name": "Mark Wilson",

"address": "654 Cedar St, City",

"phone": "9876543214",

"email": "mark.wilson@example.com",

"dept": "IT"

}

]

employee\_collection.insert\_many(employees)

print("Inserted 5 documents into the employee collection.")

# ii. Display all employees in "Accounts" department

accounts\_employees = employee\_collection.find({"dept": "Accounts"})

print("Employees in the 'Accounts' department:")

for employee in accounts\_employees:

print(employee)

# iii. Delete employee with ID - 210345

employee\_collection.delete\_one({"ID": 210345})

print("Employee with ID 210345 deleted.")

# iv. Update phone with new phone for employee ID- 123

employee\_collection.update\_one({"ID": 123}, {"$set": {"phone": "9999999999"}})

print("Phone number updated for employee with ID 123.")

# Close the MongoDB connection

client.close()

1. Write a Python program to create "movie" collection with fields (movie\_id, movie\_name, movie\_type, movie\_year, movie\_lead\_actor, movie\_director) in MongoDB. Perform the following operations-

i. Insert minimum 5 documents in movie collection.

ii. Display the list of all movies released in year 2020.

iii. Update the movie \_type to "Comedy" wherever movie\_lead\_actor is "Govinda".

iv. Display the list of movies in descending order of movie\_year.

v. Delete movie having movie\_director as "David Dhavan".

Ans:

pip install pymongo

import pymongo

client=pymongo.MongoClient("mongodb+srv://sgausiya28:root@cluster0.czdcv6d.mongodb.net/?retryWrites=true&w=majority")

db=client['db1']

employee\_collection=db['emp']

# i. Insert minimum 5 documents in movie collection

movies = [

{

"movie\_id": 1,

"movie\_name": "Movie A",

"movie\_type": "Drama",

"movie\_year": 2020,

"movie\_lead\_actor": "Actor A",

"movie\_director": "Director A"

},

{

"movie\_id": 2,

"movie\_name": "Movie B",

"movie\_type": "Comedy",

"movie\_year": 2020,

"movie\_lead\_actor": "Actor B",

"movie\_director": "Director B"

},

{

"movie\_id": 3,

"movie\_name": "Movie C",

"movie\_type": "Action",

"movie\_year": 2021,

"movie\_lead\_actor": "Actor C",

"movie\_director": "Director C"

},

{

"movie\_id": 4,

"movie\_name": "Movie D",

"movie\_type": "Comedy",

"movie\_year": 2019,

"movie\_lead\_actor": "Actor D",

"movie\_director": "Director D"

},

{

"movie\_id": 5,

"movie\_name": "Movie E",

"movie\_type": "Drama",

"movie\_year": 2022,

"movie\_lead\_actor": "Actor E",

"movie\_director": "Director E"

}

]

movie\_collection.insert\_many(movies)

print("Inserted 5 documents into the movie collection.")

# ii. Display the list of all movies released in year 2020

movies\_2020 = movie\_collection.find({"movie\_year": 2020})

print("Movies released in 2020:")

for movie in movies\_2020:

print(movie)

# iii. Update the movie\_type to "Comedy" wherever movie\_lead\_actor is "Govinda"

movie\_collection.update\_many({"movie\_lead\_actor": "Govinda"}, {"$set": {"movie\_type": "Comedy"}})

print("Updated movie\_type to 'Comedy' for movies with movie\_lead\_actor as 'Govinda'.")

# iv. Display the list of movies in descending order of movie\_year

movies\_descending\_order = movie\_collection.find().sort("movie\_year", pymongo.DESCENDING)

print("Movies in descending order of movie\_year:")

for movie in movies\_descending\_order:

print(movie)

# v. Delete movie having movie\_director as "David Dhavan"

movie\_collection.delete\_one({"movie\_director": "David Dhavan"})

print("Deleted movie with movie\_director 'David Dhavan'.")

# Close the MongoDB connection

client.close()

1. Write a Python program to create "Library" database. In Library database create 2 collections such as "Book" with fields (book\_id, title, author, publisher, price, copies) and "Transaction" with fields (trans\_id, trans\_type, trans\_date, stud\_id, book\_id). Perform the following operations -

i. Insert minimum 5 documents in Book & Transaction collection.

ii. Display the list of books in Library in ascedning order of Title.

iii. Display the trasaction details for the transactions done on "18-06-2023".

iv. Update the copies of books to 10 for the subject of Java.

v. Delete the trasactions done in year 2020.

Ans:

pip install pymongo

import pymongo

client=pymongo.MongoClient("mongodb+srv://sgausiya28:root@cluster0.czdcv6d.mongodb.net/?retryWrites=true&w=majority")

db=client['db1']

employee\_collection=db['emp']

book\_collection = db["Book"]

# Create or access the "Transaction" collection

transaction\_collection = db["Transaction"]

# i. Insert minimum 5 documents in Book & Transaction collection

books = [

{

"book\_id": 1,

"title": "Book A",

"author": "Author A",

"publisher": "Publisher A",

"price": 20.99,

"copies": 5

},

{

"book\_id": 2,

"title": "Book B",

"author": "Author B",

"publisher": "Publisher B",

"price": 15.99,

"copies": 3

},

{

"book\_id": 3,

"title": "Book C",

"author": "Author C",

"publisher": "Publisher C",

"price": 12.99,

"copies": 8

},

{

"book\_id": 4,

"title": "Book D",

"author": "Author D",

"publisher": "Publisher D",

"price": 18.99,

"copies": 10

},

{

"book\_id": 5,

"title": "Book E",

"author": "Author E",

"publisher": "Publisher E",

"price": 25.99,

"copies": 2

}

]

book\_collection.insert\_many(books)

print("Inserted 5 documents into the Book collection.")

transactions = [

{

"trans\_id": 1,

"trans\_type": "Borrow",

"trans\_date": "18-06-2023",

"stud\_id": 101,

"book\_id": 1

},

{

"trans\_id": 2,

"trans\_type": "Return",

"trans\_date": "18-06-2023",

"stud\_id": 102,

"book\_id": 3

},

{

"trans\_id": 3,

"trans\_type": "Borrow",

"trans\_date": "20-06-2023",

"stud\_id": 103,

"book\_id": 2

},

{

"trans\_id": 4,

"trans\_type": "Return",

"trans\_date": "22-06-2023",

"stud\_id": 104,

"book\_id": 4

},

{

"trans\_id": 5,

"trans\_type": "Borrow",

"trans\_date": "25-06-2023",

"stud\_id": 105,

"book\_id": 5

}

]

transaction\_collection.insert\_many(transactions)

print("Inserted 5 documents into the Transaction collection.")

# ii. Display the list of books in Library in ascending order of Title

books\_asc\_order = book\_collection.find().sort("title", pymongo.ASCENDING)

print("Books in ascending order of Title:")

for book in books\_asc\_order:

print(book)

# iii. Display the transaction details for the transactions done on "18-06-2023"

transactions\_18\_june = transaction\_collection.find({"trans\_date": "18-06-2023"})

print("Transactions done on 18-06-2023:")

for transaction in transactions\_18\_june:

print(transaction)

# iv. Update the copies of books to 10 for the subject of Java

book\_collection.update\_many({"title": "Book A"}, {"$set": {"copies": 10}})

print("Updated copies of books with title 'Book A' to 10.")

# v. Delete the transactions done in year 2020

transaction\_collection.delete\_many({"trans\_date": {"$regex": ".\*2020.\*"}})

print("Deleted transactions done in the year 2020.")

# Close the MongoDB connection

client.close()

1. Write a menu driven(add subtract, multiply and division, exit) program to handle two 1 D array

Ans

def add\_arrays(arr1, arr2):

return [x + y for x, y in zip(arr1, arr2)]

def subtract\_arrays(arr1, arr2):

return [x - y for x, y in zip(arr1, arr2)]

def multiply\_arrays(arr1, arr2):

return [x \* y for x, y in zip(arr1, arr2)]

def divide\_arrays(arr1, arr2):

return [x / y for x, y in zip(arr1, arr2)]

def display\_array(arr):

print(arr)

def menu\_1d():

arr1 = []

arr2 = []

while True:

print("1. Add arrays")

print("2. Subtract arrays")

print("3. Multiply arrays")

print("4. Divide arrays")

print("5. Display array 1")

print("6. Display array 2")

print("7. Exit")

choice = int(input("Enter your choice: "))

if choice == 1:

result = add\_arrays(arr1, arr2)

print("Result: ", result)

elif choice == 2:

result = subtract\_arrays(arr1, arr2)

print("Result: ", result)

elif choice == 3:

result = multiply\_arrays(arr1, arr2)

print("Result: ", result)

elif choice == 4:

result = divide\_arrays(arr1, arr2)

print("Result: ", result)

elif choice == 5:

display\_array(arr1)

elif choice == 6:

display\_array(arr2)

elif choice == 7:

break

else:

print("Invalid choice. Try again.")

menu\_1d()