|  |  |
| --- | --- |
| **ffSR. NO.** | **NAME OF EXPERIMENT** |
| 1a | Write python programs to accept price of an item, percentage of discount and then display final value of on screen.  Ans;  def calculate\_discounted\_price(price, discount\_percentage):  discount\_amount = price \* (discount\_percentage / 100)  discounted\_price = price - discount\_amount  return discounted\_price    def main():  try:  price = float(input("Enter the price of the item: $"))  discount\_percentage = float(input("Enter the discount percentage: "))    if discount\_percentage < 0 or discount\_percentage > 100:  print("Discount percentage must be between 0 and 100.")  return    discounted\_price = calculate\_discounted\_price(price, discount\_percentage)    print(f"The final discounted price is: Rs{discounted\_price:.2f}")  except ValueError:  print("Invalid input! Please enter numeric values.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 1b | Write python programs to demonstrate basic maths operation.  Ans:  def add(a, b):  return a + b    def subtract(a, b):  return a - b    def multiply(a, b):  return a \* b    def divide(a, b):  if b != 0:  return a / b  else:  return "Error: Division by zero!"    def main():  num1 = float(input("Enter the first number: "))  num2 = float(input("Enter the second number: "))    print("1. Addition")  print("2. Subtraction")  print("3. Multiplication")  print("4. Division")    choice = input("Enter your choice (1/2/3/4): ")    if choice == '1':  print("Result:", add(num1, num2))  elif choice == '2':  print("Result:", subtract(num1, num2))  elif choice == '3':  print("Result:", multiply(num1, num2))  elif choice == '4':  print("Result:", divide(num1, num2))  else:  print("Invalid choice!")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 2a | Write a Python program to count the number of even and odd numbers from a series of numbers.  Ans:  def count\_even\_odd(numbers):  even\_count = 0  odd\_count = 0    for num in numbers:  if num % 2 == 0:  even\_count += 1  else:  odd\_count += 1    return even\_count, odd\_count    def main():  try:  series = input("Enter a series of numbers separated by spaces: ")  numbers = list(map(int, series.split()))    even\_count, odd\_count = count\_even\_odd(numbers)    print("Number of even numbers:", even\_count)  print("Number of odd numbers:", odd\_count)  except ValueError:  print("Invalid input! Please enter numeric values separated by spaces.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 2b | Write a Python program to convert temperatures to and from Celsius, Fahrenheit.  Ans:  def celsius\_to\_fahrenheit(celsius):  return (celsius \* 9/5) + 32    def fahrenheit\_to\_celsius(fahrenheit):  return (fahrenheit - 32) \* 5/9    def main():  try:  choice = input("Enter '1' to convert from Celsius to Fahrenheit or '2' to convert from Fahrenheit to Celsius: ")    if choice == '1':  celsius = float(input("Enter temperature in Celsius: "))  fahrenheit = celsius\_to\_fahrenheit(celsius)  print(f"{celsius:.2f} Celsius is equal to {fahrenheit:.2f} Fahrenheit.")  elif choice == '2':  fahrenheit = float(input("Enter temperature in Fahrenheit: "))  celsius = fahrenheit\_to\_celsius(fahrenheit)  print(f"{fahrenheit:.2f} Fahrenheit is equal to {celsius:.2f} Celsius.")  else:  print("Invalid choice!")  except ValueError:  print("Invalid input! Please enter a numeric value for temperature.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 3a | Write a python program to accept marks of five subjects from student. Calculate percentage and display result like First Class, Second Class, Pass, Fail using decision making statements.  Ans:  def calculate\_percentage(subject\_marks):  total\_marks = sum(subject\_marks)  percentage = (total\_marks / (len(subject\_marks) \* 100)) \* 100  return percentage    def determine\_result(percentage):  if percentage >= 60:  return "First Class"  elif percentage >= 50:  return "Second Class"  elif percentage >= 40:  return "Pass"  else:  return "Fail"    def main():  try:  subject\_marks = []  for i in range(5):  mark = float(input(f"Enter marks of subject {i+1}: "))  subject\_marks.append(mark)    percentage = calculate\_percentage(subject\_marks)  result = determine\_result(percentage)    print(f"Percentage: {percentage:.2f}%")  print("Result:", result)  except ValueError:  print("Invalid input! Please enter numeric values for marks.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 3b | Write a Python program to print factorial of entered number.  Ans:  def factorial(n):  if n == 0:  return 1  else:  return n \* factorial(n - 1)    def main():  try:  num = int(input("Enter a number to find its factorial: "))  if num < 0:  print("Factorial is not defined for negative numbers.")  else:  result = factorial(num)  print(f"The factorial of {num} is: {result}")  except ValueError:  print("Invalid input! Please enter an integer.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 4a | Write a Python program to print the numbers of a specified list after removing even numbers from it.  Ans :  def remove\_even\_numbers(lst):  return [num for num in lst if num % 2 != 0]    def main():  try:  # Input list of numbers  numbers = input("Enter a list of numbers separated by spaces: ")  numbers1 = list(map(int, numbers.split()))    # Remove even numbers from the list  filtered\_numbers = remove\_even\_numbers(numbers1)    # Print the filtered list  print("List after removing even numbers:", filtered\_numbers)  except ValueError:  print("Invalid input! Please enter valid integers separated by spaces.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 4b | Write a Python program to check whether an element exists within a tuple.  Ans:  def element\_exists\_in\_tuple(element, tup):  return element in tup    def main():  try:  # Input tuple  tup = tuple(input("Enter elements of the tuple separated by spaces: ").split())    # Input element to check  element = input("Enter the element to check in the tuple: ")    # Check if the element exists in the tuple  if element\_exists\_in\_tuple(element, tup):  print(f"The element '{element}' exists in the tuple.")  else:  print(f"The element '{element}' does not exist in the tuple.")  except ValueError:  print("Invalid input! Please enter valid elements separated by spaces.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| Q 5a | Write a Python program to concatenate following dictionaries to create a new one. dict1={1:10,2:20}  dict2={3:30,4:40}  dict3={5:50,6:60}  Result={1:10,2:20,3:30,4:40,5:50,6:60}  Ans :  def concatenate\_dicts(\*dicts):  result = {}  for d in dicts:  result.update(d)  return result    def main():  dict1 = {1: 10, 2: 20}  dict2 = {3: 30, 4: 40}  dict3 = {5: 50, 6: 60}    concatenated\_dict = concatenate\_dicts(dict1, dict2, dict3)  print("Result:", concatenated\_dict)    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 5b | Write a Python program to check whether an element exists within a List.  Ans:  def element\_exists\_in\_list(element, lst):  return element in lst    def main():  try:  # Input list  lst = input("Enter elements of the list separated by spaces: ").split()    # Input element to check  element = input("Enter the element to check in the list: ")    # Check if the element exists in the list  if element\_exists\_in\_list(element, lst):  print(f"The element '{element}' exists in the list.")  else:  print(f"The element '{element}' does not exist in the list.")  except ValueError:  print("Invalid input! Please enter valid elements separated by spaces.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 6a | Write a Python program to accept a number and check if its odd or even.  Ans :  def check\_odd\_even(number):  if number % 2 == 0:  return "even"  else:  return "odd"    def main():  try:  number = int(input("Enter a number: "))  result = check\_odd\_even(number)  print(f"The number {number} is {result}.")  except ValueError:  print("Invalid input! Please enter an integer.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 6b | Write python program to check entered number is prime or not.  Ans:  def is\_prime(number):  if number <= 1:  return False    for i in range(2, number):  if number % i == 0:  return False    return True  def main():  num = int(input("Enter a number: "))  if is\_prime(num):  print(f"{num} is a prime number.")  else:  print(f"{num} is not a prime number.")  if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 7 | Write a python program to design basic calculator to perform addition, subtraction, multiplication and division using function.  Ans:  def add(num1, num2):  return num1 + num2    def subtract(num1, num2):  return num1 - num2    def multiply(num1, num2):  return num1 \* num2    def divide(num1, num2):  if num2 != 0:  return num1 / num2  else:  return "Error: Division by zero!"    def main():  try:  num1 = float(input("Enter the first number: "))  num2 = float(input("Enter the second number: "))    print("1. Addition")  print("2. Subtraction")  print("3. Multiplication")  print("4. Division")    choice = input("Enter your choice (1/2/3/4): ")    if choice == '1':  print("Result:", add(num1, num2))  elif choice == '2':  print("Result:", subtract(num1, num2))  elif choice == '3':  print("Result:", multiply(num1, num2))  elif choice == '4':  print("Result:", divide(num1, num2))  else:  print("Invalid choice!")  except ValueError:  print("Invalid input! Please enter numeric values.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 8 | Write a python program to demonstrate multiple inheritance.  Ans:  class Parent1:  def method1(self):  print("Method 1 from Parent1")    class Parent2:  def method2(self):  print("Method 2 from Parent2")    class Child(Parent1, Parent2):  def method3(self):  print("Method 3 from Child")    def main():  obj = Child()  obj.method1() # Accessing method from Parent1  obj.method2() # Accessing method from Parent2  obj.method3() # Accessing method from Child    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 9 | Write python program to demonstrate method overriding  Ans:  class Parent:  def show(self):  print("Parent's show method")    class Child(Parent):  def show(self):  print("Child's show method")    def main():  parent\_obj = Parent()  child\_obj = Child()    parent\_obj.show() # Calls Parent's show method  child\_obj.show() # Calls Child's show method (overridden)    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 10 | Write a python program to calculate area of i) Circle, ii) Square and iii) Rectangle using abstract class.  Ans:  from abc import ABC, abstractmethod  import math    class Shape(ABC):  @abstractmethod  def calculate\_area(self):  pass    class Circle(Shape):  def \_\_init\_\_(self, radius):  self.radius = radius    def calculate\_area(self):  return math.pi \* self.radius \*\* 2    class Square(Shape):  def \_\_init\_\_(self, side):  self.side = side    def calculate\_area(self):  return self.side \*\* 2    class Rectangle(Shape):  def \_\_init\_\_(self, length, width):  self.length = length  self.width = width    def calculate\_area(self):  return self.length \* self.width    def main():  circle = Circle(5)  print("Area of circle:", circle.calculate\_area())    square = Square(4)  print("Area of square:", square.calculate\_area())    rectangle = Rectangle(3, 6)  print("Area of rectangle:", rectangle.calculate\_area())    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 11 | Write a python program to calculate area of i) Circle, ii) Square and iii) Rectangle using interface.  Ans:  from abc import ABC, abstractmethod  import math    class Shape(ABC):  @abstractmethod  def calculate\_area(self):  pass    class Circle(Shape):  def \_\_init\_\_(self, radius):  self.radius = radius    def calculate\_area(self):  return math.pi \* self.radius \*\* 2    class Square(Shape):  def \_\_init\_\_(self, side):  self.side = side    def calculate\_area(self):  return self.side \*\* 2    class Rectangle(Shape):  def \_\_init\_\_(self, length, width):  self.length = length  self.width = width    def calculate\_area(self):  return self.length \* self.width    def main():  try:  shape\_choice = input("Enter 'circle', 'square', or 'rectangle' to calculate the area: ").lower()    if shape\_choice == 'circle':  radius = float(input("Enter the radius of the circle: "))  shape = Circle(radius)  elif shape\_choice == 'square':  side = float(input("Enter the length of a side of the square: "))  shape = Square(side)  elif shape\_choice == 'rectangle':  length = float(input("Enter the length of the rectangle: "))  width = float(input("Enter the width of the rectangle: "))  shape = Rectangle(length, width)  else:  print("Invalid choice!")  return    area = shape.calculate\_area()  print(f"Area of {shape\_choice}: {area:.2f}")  except ValueError:  print("Invalid input! Please enter numeric values.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 12 | Write python programs to understand creation of a menu driven application which should cover all the built-in exceptions in python.  Ans:  def divide\_numbers():  try:  num1 = float(input("Enter the first number: "))  num2 = float(input("Enter the second number: "))  result = num1 / num2  print("Result of division:", result)  except ZeroDivisionError:  print("Error: Division by zero!")  except ValueError:  print("Error: Invalid input! Please enter numeric values.")    def index\_error\_demo():  try:  my\_list = [1, 2, 3]  index = int(input("Enter the index to access: "))  print("Element at index", index, "is", my\_list[index])  except IndexError:  print("Error: Index out of range!")    def key\_error\_demo():  try:  my\_dict = {"a": 1, "b": 2, "c": 3}  key = input("Enter the key to access: ")  print("Value associated with key", key, "is", my\_dict[key])  except KeyError:  print("Error: Key not found!")    def file\_open\_demo():  try:  filename = input("Enter the filename to open: ")  with open(filename, "r") as file:  contents = file.read()  print("File contents:\n", contents)  except FileNotFoundError:  print("Error: File not found!")    def main():  while True:  print("\nMenu:")  print("1. Divide Numbers")  print("2. Index Error Demo")  print("3. Key Error Demo")  print("4. File Open Demo")  print("5. Exit")    choice = input("Enter your choice: ")    if choice == '1':  divide\_numbers()  elif choice == '2':  index\_error\_demo()  elif choice == '3':  key\_error\_demo()  elif choice == '4':  file\_open\_demo()  elif choice == '5':  print("Exiting...")  break  else:  print("Invalid choice! Please enter a valid option.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 13 | Write python program to create user defined packages and import them in program.  Ans:  Addition.py  def add(a, b):  return a + b  Substraction.py  def sub(a, b):  return a – b  multiplication.py  def mul(a, b):  return a \* b  Division.py  def mul(a, b):  return a /b  Main.py  from calculator.addition import add  from calculator.subtraction import sub  from calculator.multiplication import mul  from calculator.division import div    print("Addition of 8 and 4 is:", add(8, 4))  print("Subtraction of 8 and 4 is:", sub(8, 4))  print("Multiplication of 8 and 4 is:", mul(8, 4))  print("Division of 8 and 4 is:", div(8, 4)) |
| 14 | Write python program to understand different File Handling operations in Python.  Ans:  def write\_to\_file(filename):  try:  with open(filename, "w") as file:  content = input("Enter content to write to the file: ")  file.write(content)  print("Content written to the file successfully.")  except IOError:  print("Error: Could not write to the file.")    def read\_from\_file(filename):  try:  with open(filename, "r") as file:  content = file.read()  print("Content read from the file:")  print(content)  except FileNotFoundError:  print("Error: File not found.")  except IOError:  print("Error: Could not read from the file.")    def append\_to\_file(filename):  try:  with open(filename, "a") as file:  content = input("Enter content to append to the file: ")  file.write("\n" + content)  print("Content appended to the file successfully.")  except IOError:  print("Error: Could not append to the file.")    def main():  filename = "example.txt"    print("1. Write to file")  print("2. Read from file")  print("3. Append to file")  print("4. Exit")    while True:  choice = input("Enter your choice (1/2/3/4): ")    if choice == '1':  write\_to\_file(filename)  elif choice == '2':  read\_from\_file(filename)  elif choice == '3':  append\_to\_file(filename)  elif choice == '4':  print("Exiting...")  break  else:  print("Invalid choice! Please enter a valid option.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 15 | Write python programs to understand designing Graphical user interface (GUI) using Tkinter.  Ans:  from tkinter import \*  top = Tk()  top.title("SE-IT 17104074")  c = Canvas(top,bg="blue",height=1000,width=1000)  fnt = ('Times','22','bold italic underline')  text = c.create\_text(250,40,text="My First Tkinter Application on canvas",  font=fnt,fill="red",activefill="cyan3")  line = c.create\_line(80,80,200,80,200,200,width=6,fill="cyan3")line = c.create\_line(80,80,300,80,300,300,width=6,fill="cyan3")oval = c.create\_oval(120,120,400,300,width=6,fill="red",outline="green",activefill=3")  poly = c.create\_polygon(320,320,320,420,420,320,width=6,fill="yellow",outline="gactivefill="cyan3")  rect = c.create\_rectangle(360,460,550,550,width=6,fill="green",outline="green",ac="cyan3")  arc =  c.create\_arc(500,100,600,300,start=0,extent=180,width=6,outline="green"arc") file1=PhotoImage(file="python.png",height=300,width=300) file2=PhotoImage(file="Java.png",height=300,width=300) id=c.create\_image(300,400,anchor=NE,image=file1,activeimage=file2)  c.pack() top.mainloop() |
| 16 | Write python programs to perform CRUD operations in python using SQLite.  Ans:  import sqlite3  conn=sqlite3.connect('example.db')  print ("Opened database successfully"); conn.execute('''CREATE TABLE EMPLOYEE11 (FIRST\_NAME CHAR(20) NOT NULL, LAST\_NAME CHAR(20), AGE INT,SEX CHAR(1),INCOME FLOAT )''')  print ("Table created successfully");  conn.execute("INSERT INTO EMPLOYEE11(FIRST\_NAME,LAST\_NAME, AGE, SEX, INCOME)VALUES ('RAJESH','GANGULY',28,'M',93776)"); conn.execute("INSERT INTO EMPLOYEE11(FIRST\_NAME,LAST\_NAME, AGE, SEX, INCOME) VALUES ('SACHIN','TENDULKAR',40,'M',98463)"); conn.execute("INSERT INTO EMPLOYEE11(FIRST\_NAME,LAST\_NAME, AGE, SEX, INCOME)VALUES ('ANIL','KUMBLE',42,'M',64300)"); conn.execute("INSERT INTO EMPLOYEE11(FIRST\_NAME,LAST\_NAME, AGE, SEX, INCOME)VALUES ('RAHUL','DRAVID',39,'M',84300)"); conn.commit()  print ("Records inserted successfully");  data = conn.execute("select \* from EMPLOYEE11");  for row in data:  print ("FIRST\_NAME = ", row[0])  print ("LAST\_NAME = ", row[1])  print ("AGE = ", row[2])  print ("SEX = ", row[3])  print ("INCOME = ", row[4])  conn.execute("UPDATE EMPLOYEE11 SET AGE = AGE + 1 WHERE SEX = '%c'" % ('M'))  conn.commit()  data = conn.execute("select \* from EMPLOYEE11"); print("Updated table");  for row in data:  print ("FIRST\_NAME = ", row[0])  print ("LAST\_NAME = ", row[1])  print ("AGE = ", row[2])  print ("SEX = ", row[3])  print ("INCOME = ", row[4])  conn.execute("DELETE FROM EMPLOYEE11 WHERE AGE > '%d'" % (40))  conn.commit()  data = conn.execute("select \* from EMPLOYEE11"); print("After deletion")  for row in data:  print ("FIRST\_NAME = ", row[0])  print ("LAST\_NAME = ", row[1])  print ("AGE = ", row[2])  print ("SEX = ", row[3])  print ("INCOME = ", row[4])  conn.close() |
| 17 | Write a python program to demonstrate user defined exception.  Ans:  class MyCustomException(Exception):  def \_\_init\_\_(self, message):  self.message = message    def divide\_numbers(num1, num2):  if num2 == 0:  raise MyCustomException("Division by zero is not allowed")  else:  return num1 / num2    def main():  try:  num1 = float(input("Enter the first number: "))  num2 = float(input("Enter the second number: "))  result = divide\_numbers(num1, num2)  print("Result of division:", result)  except ValueError:  print("Invalid input! Please enter numeric values.")  except MyCustomException as e:  print("Error:", e.message)    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 18a | Write a python program to check largest number among three.  Ans:  def find\_largest(num1, num2, num3):  if num1 >= num2 and num1 >= num3:  return num1  elif num2 >= num1 and num2 >= num3:  return num2  else:  return num3    def main():  try:  num1 = float(input("Enter the first number: "))  num2 = float(input("Enter the second number: "))  num3 = float(input("Enter the third number: "))    largest = find\_largest(num1, num2, num3)  print("The largest number is:", largest)  except ValueError:  print("Invalid input! Please enter numeric values.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 18b | Write a python program to check entered number is armstrong number.  Ans:  def is\_armstrong\_number(number):  num\_str = str(number)  num\_digits = len(num\_str)  armstrong\_sum = 0    for digit in num\_str:  digit\_value = int(digit)  powered\_digit = digit\_value \*\* num\_digits  armstrong\_sum += powered\_digit    return armstrong\_sum == number  def main():  num = int(input("Enter a number: "))  if is\_armstrong\_number(num):  print(f"{num} is an Armstrong number.")  else:  print(f"{num} is not an Armstrong number.")  if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 19 | Write a python program to print nth Fibonacci numbers.  Ans:  def fibonacci\_numbers(n):  fib\_series = []  a, b = 0, 1  for \_ in range(n):  fib\_series.append(a)  a, b = b, a + b    return fib\_series    def main():  n = int(input("Enter the number of Fibonacci numbers to print: "))  fib\_list = fibonacci\_numbers(n)  print(f"Fibonacci series of {n} numbers:")  print(fib\_list)  if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 20a | Python Program for Sum of squares of first n natural numbers  Ans:  def sum\_of\_squares(n):  return sum(i\*\*2 for i in range(1, n+1))    def main():  try:  n = int(input("Enter the value of n: "))  if n <= 0:  print("Please enter a positive integer.")  else:  result = sum\_of\_squares(n)  print("The sum of squares of the first", n, "natural numbers is:", result)  except ValueError:  print("Invalid input! Please enter an integer.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |
| 20b | Python Program for cube sum of first n natural numbers.  Ans:  def cube\_sum(n):  return sum(i\*\*3 for i in range(1, n+1))    def main():  try:  n = int(input("Enter the value of n: "))  if n <= 0:  print("Please enter a positive integer.")  else:  result = cube\_sum(n)  print("The cube sum of the first", n, "natural numbers is:", result)  except ValueError:  print("Invalid input! Please enter an integer.")    if \_\_name\_\_ == "\_\_main\_\_":  main() |