**SRM Institute of Science and Technology**

**Ramapuram Chennai**

**Faculty of Science & Humanities**

**(A Place for Transformation)**

**Department of Computer Science and Applications (BCA & BCA-GenAI)**

PRACTICAL RECORD

**NAME :**

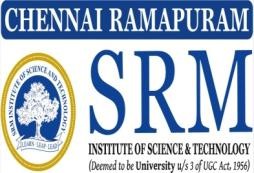
**REGISTER NUMBER :**

**COURSE : BCA**

**SEMESTER / YEAR : V / III SUBJECT CODE : UCA23501J -**

**SUBJECTNAME : PYTHON PROGRAMMING**

**October 2025**

**SRM Institute of Science and Technology**

**Ramapuram Chennai**

**Faculty of Science & Humanities**

**(A Place for Transformation)**

**Department of Computer Science and Applications (BCA & BCA-GenAI)**

**REGISTERNUMBER:**

**BONAFIDECERTIFICATE**

This is to certify that the bonafide work done by in the subject **UCA23501J - PYTHON PROGRAMMING** at, SRM Institute of Science and Technology, Ramapuram Chennai in **October 2025**.

**STAFFIN-CHARGE HEAD OF THE DEPARTMENT**

Submitted for the University Practical Examination held at SRM Institute of Science and Technology, Ramapuram Chennai on .

**INTERNAL EXAMINER 1 INTERNAL EXAMINER 2**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ex.No** | **Date** | **Description** | **Page**  **No** | **Signature** |
| 1 |  | Write a Python code to display system information using pywhois. | 1 |  |
| 2 |  | The Magic 8 Ball is a toy used for fortune-telling or seeking advice. | 2 |  |
| 3 |  | Check whether a number is prime or not | 4 |  |
| 4 |  | Make a Simple Calculator | 5 |  |
| 5 |  | Find the Factorial of a Number | 7 |  |
| 6 |  | Python Program to Generate a Random Number | 8 |  |
| 7 |  | Python Program to Display the multiplication Table | 9 |  |
| 8 |  | Python Program to Convert Decimal to Binary, Octal and Hexadecimal | 10 |  |
| 9 |  | Python Program to Transpose a Matrix | 11 |  |
| 10 |  | Python Program to Multiply Two Matrices | 12 |  |
| 11 |  | Python Program to Check Whether a String is Palindrome or Not | 13 |  |
| 12 |  | Python Program to Sort Words in Alphabetic Order | 14 |  |
| 13 |  | Python Program for Inheritance. | 15 |  |
| 14 |  | Python Program for Operator overloading. | 16 |  |
| 15 |  | Python Program for Exception Handling. | 17 |  |
| 16 |  | Program to read and write text and numbers | 18 |  |
| 17 |  | Using a List to Find the Median of a Set of Numbers  Program using sorting and searching | 20 |  |
| 18 |  | When the user enters a statement, the program responds in one of two ways: 1 With a randomly chosen hedge, such as “Please tell me more.” 2 By  changing some key words in the user’s input string and appending this string to a randomly chosen qualifier.  Thus, to “My teacher always plays favorites,” the  program might reply, “Why do you say that your teacher always plays favorites?” | 22 |  |
| 19 |  | Program using classes and methods | 25 |  |
| 20 |  | Program using polymorphism, abstract classes | 27 |  |

**EX.No. : 1 RegNo:**

**Date : Name :**

**Write a Python code to display system information using pywhois.**

# AIM:

To Write a Python code to display system information using pywhois

# PROGRAM:

import platform

my\_system = platform.uname() print(f"System: {my\_system.system}") print(f"Node Name: {my\_system.node}") print(f"Release: {my\_system.release}") print(f"Version: {my\_system.version}") print(f"Machine: {my\_system.machine}") print(f"Processor: {my\_system.processor}")

# OUTPUT:

System: Windows

Node Name: DESKTOP-7ILF0RN

Release: 10

Version: 10.0.19041 Machine: AMD64

Processor: Intel64 Family 6 Model 165 Stepping 5, GenuineIntel

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 2 RegNo:**

**Date : Name :**

**The Magic 8 Ball is a toy used for fortune-telling or seeking advice.**

# AIM:

To a python program to generate the Magic 8 Ball is for fortune-telling or seeking advice.

# PROGRAM:

import random

answers = ['It is certain', 'It is decidedly so', 'Without a doubt', 'Yes –

definitely', 'You may rely on it', 'As I see it, yes', 'Most likely', 'Outlook good', 'Yes Signs point to yes ', 'Reply hazy', 'try again', 'Ask again later', 'Better not tell you now', 'Cannot predict now', 'Concentrat e and ask again', 'Dont count on it', 'My reply is no', 'My sources say no', 'Outlook not so good', 'Very doubtful']

print('Hello World, I am the Magic 8 Ball, What is your name?') name = input()

print('hello ' + name)

def Magic8Ball():

print('Ask me a question.') input()

print (answers[random.randint(0, len(answers)-1)] ) print('I hope that helped!')

Replay()

def Replay():

print ('Do you have another question? [Y/N] ') reply = input()

if reply == 'Y': Magic8Ball()

elif reply == 'N': #exit() print('Bye')

else:

print('I apologies, I did not catch that. Please repeat.') Replay()

Magic8Ball()

# OUTPUT:

Hello World, I am the Magic 8 Ball, What is your name? joe

hello joe

Ask me a question.

will i pass this semester ? Very doubtful

I hope that helped!

Do you have another question? [Y/N] N

Bye

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 3 RegNo:**

**Date : Name :**

**Check whether a number is prime or not**

# AIM:

To write a program to check whether a number is prime or not in python.

# PROGRAM:

num = int(input("Enter a number: ")) if num > 1:

for i in range(2,num): if (num % i) == 0:

print(num,"is not a prime number") print(i,"times",num//i,"is",num) break

else:

print(num,"is a prime number")

else:

print(num,"is not a prime number")

# OUTPUT:

Enter a number: 5

1. is a prime number

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 4 RegNo:**

**Date : Name :**

**Make a Simple Calculator**

# AIM:

Python Program to Make a Simple Calculator.

# PROGRAM:

def add(x, y): return x + y

def subtract(x, y): return x - y

def multiply(x, y): return x \* y

def divide(x, y): return x / y

print("Select operation.") print("1.Add") print("2.Subtract") print("3.Multiply") print("4.Divide")

while True:

choice = input("Enter choice(1/2/3/4): ")

if choice in ('1', '2', '3', '4'):

num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: "))

if choice == '1':

print(num1, "+", num2, "=", add(num1, num2))

elif choice == '2':

print(num1, "-", num2, "=", subtract(num1, num2))

elif choice == '3':

print(num1, "\*", num2, "=", multiply(num1, num2))

elif choice == '4':

print(num1, "/", num2, "=", divide(num1, num2)) break

else:

print("Invalid Input")

# OUTPUT:

Select operation. 1.Add 2.Subtract 3.Multiply 4.Divide

Enter choice(1/2/3/4): 1 Enter first number: 23 Enter second number: 45

23.0 + 45.0 = 68.0

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 5 RegNo:**

**Date : Name :**

**Find the Factorial of a Number**

# AIM:

To write a python program for Finding the Factorial of a Number.

# PROGRAM:

num = int(input("Enter a number: ")) factorial = 1

# check if the number is negative, positive or zero if num < 0:

print("Sorry, factorial does not exist for negative numbers") elif num == 0:

print("The factorial of 0 is 1") else:

for i in range(1,num + 1): factorial = factorial\*i

print("The factorial of",num,"is",factorial)

# OUTPUT:

Enter a number: 5

The factorial of 5 is 120

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 6 RegNo:**

**Date : Name :**

**Python Program to Generate a Random Number**

# AIM:

To write a python Program for generating a Random Number

# PROGRAM:

import random

x=int(input("Enter a number to Predict ")) y=random.randint(0,9)#single

if x==y:

print("Exact Prediction ",x) elif (x-y) ==1 :

print("Ohhh!Missed By One ", y) else:

print("Prediction dose not match ",y )

# OUTPUT:

Enter a number to Predict 7 Prediction dose not match 58

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 7 RegNo:**

**Date : Name :**

**Python Program to Display the multiplication Table**

# AIM:

TO write a Python Program for Displaying the multiplication Table.

# PROGRAM:

x=int(input("Enter a number to generate a table")) for i in range(1,11):

print("{0} \* {1} = {2}".format(i,x,i\*x))

# OUTPUT:

Enter a number to generate a table 5

1 \* 5 = 5

2 \* 5 = 10

3 \* 5 = 15

4 \* 5 = 20

5 \* 5 = 25

1. \* 5 = 30
2. \* 5 = 35
3. \* 5 = 40
4. \* 5 = 45

10 \* 5 = 50

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 8 RegNo:**

**Date : Name :**

**Python Program to Convert Decimal to Binary, Octal and Hexadecimal**

# AIM:

To Write a Python Program for Converting Decimal to Binary, Octal and Hexadecimal

# PROGRAM:

dec = int(input())

print("The decimal value Entered ", dec, "is:") print(bin(dec), "in binary.")

print(oct(dec), "in octal.") print(hex(dec), "in hexadecimal.")

# OUTPUT:

98

The decimal value Entered 98 is:

0b1100010 in binary. 0o142 in octal.

0x62 in hexadecimal.

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 9 RegNo:**

**Date : Name :**

**Python Program to Transpose a Matrix**

# AIM:

TO WRITE A Python Program FOR Transposing a Matrix

# PROGRAM:

X = [[12,7],

[4 ,5],

[3 ,8]]

result = [[0,0,0],

[0,0,0]]

# iterate through rows for i in range(len(X)):

# iterate through columns for j in range(len(X[0])):

result[j][i] = X[i][j]

for r in result:

print(r)

# OUTPUT:

[12, 4, 3]

[7, 5, 8]

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 10 RegNo:**

**Date : Name :**

# AIM:

**Python Program to Multiply Two Matrices**

To write a Python Program for Multiplying Two Matrices.

# PROGRAM:

X = [[12,7,3],

[4 ,5,6],

[7 ,8,9]]

# 3x4 matrix Y = [[5,8,1,2],

[6,7,3,0],

[4,5,9,1]]

# result is 3x4 result = [[0,0,0,0],

[0,0,0,0],

[0,0,0,0]]

for i in range(len(X)):

for j in range(len(Y[0])): for k in range(len(Y)):

result[i][j] += X[i][k] \* Y[k][j] for r in result:

print(r)

# OUTPUT:

[114, 160, 60, 27]

[74, 97, 73, 14]

[119, 157, 112, 23]

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 11 RegNo:**

**Date : Name :**

**Python Program to Check Whether a String is Palindrome or Not**

# AIM:

To write a Python Program for Checking Whether a String is Palindrome or Not

# PROGRAM:

x = "malayalam" w = ""

for i in x:

w= i + w #mal

if (x == w):

print("Yes it is palindrome") else:

print("No, it is NOT palindrome")

# OUTPUT:

Yes it is palindrome

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 12 RegNo:**

**Date : Name :**

**Python Program to Sort Words in Alphabetic Order**

# AIM:

To write a Python Program for Sorting Words in Alphabetic Order

# PROGRAM:

my\_str = input("Enter a string: ")

words = [word.lower() for word in my\_str.split()] words.sort()

print("The sorted words are:") for word in words:

print(word)

# OUTPUT:

Enter a string: I love Python an easy language to code The sorted words are:

an code easy i

language love python to

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 13 RegNo:**

**Date : Name :**

**Python Program for Inheritance.**

# AIM:

To write a program for implementing inheritance Python

# PROGRAM:

class addition:

def init (self,x,y):

self.x=x self.y=y

def addi(self,x,y):

return (x+y)

#object creation process

a=int (input("Enter a first Number: ")) b=int (input("Enter a second Number: ")) obj =addition(a,b)

print("Addition of two numbers : ",obj.addi(a,b))

# OUTPUT:

Enter a first Number: 5 Enter a second Number: 7

Addition of two numbers : 12

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 14 RegNo:**

**Date : Name :**

**Python Program for Operator overloading.**

# AIM:

Python program To implement Operator overloading. PROGRAM:

class A:

def init (self, a): self.a = a

# adding two objects def add (self, o):

return self.a + o.a

ob1 = A(1) ob2 = A(2)

ob3 = A("Python")

ob4 = A("overloading")

print(ob1 + ob2) print(ob3 + ob4)

# OUTPUT:

3

Pythonoverloading

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 15 RegNo:**

**Date : Name :**

**Python Program for Exception Handling.**

# AIM:

To write a Python Program for implementing Exception Handling.

# PROGRAM:

a=int (input("Enter a first Number: ")) b=int (input("Enter a second Number: ")) try:

k = a//b print(k)

except ZeroDivisionError:

print("Can't divide by zero") else :

print(k) finally:

print('Program closed')

# OUTPUT:

Enter a first Number: 5 Enter a second Number: 0 Can't divide by zero Program closed

# RESULT:

Thus the program has been verified and completed successfully.

**EX.No. : 16 RegNo:**

**Date : Name :**

**Program to read and write text and numbers**

**Aim:**

To write a python program to read and write text and numbers

**Program:**

f = open("sample.txt", "w") integer = 10 f.write(str(integer))

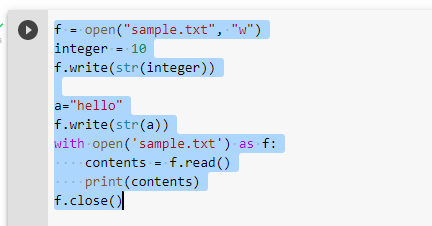
a="hello" f.write(str(a))

with open('sample.txt') as f: contents = f.read() print(contents)

f.close()

**output**

**Input:**

****

**Output:**

****

**Result:**

Thus the above program has been executed successfully.

**EX.No. : 17 RegNo:**

**Date : Name :**

**Using a List to Find the Median of a Set of Numbers Program using sorting and searching**

**Aim:**

To write a program using a List to Find the Median of a Set of Numbers

Program using sorting and searching]

**Program:**

def get\_median(ls): # sort the list

ls\_sorted = ls.sort() # find the median if len(ls) % 2 != 0:

# total number of values are odd

# subtract 1 since indexing starts at 0 m = int((len(ls)+1)/2 - 1)

return ls[m] else:

m1 = int(len(ls)/2 - 1) m2 = int(len(ls)/2) return (ls[m1]+ls[m2])/2

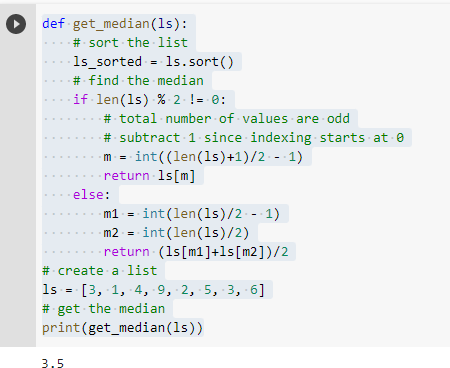
# create a list

ls = [3, 1, 4, 9, 2, 5, 3, 6]

# get the median print(get\_median(ls))

**Screenshot:**

**Input:**

****

**Output:**

****

**Result:**

Thus the above program has been executed successfully.

**EX.No. : 18 RegNo:**

**Date : Name :**

**CASE STUDY Example for Dictionary, list String Literals**

**Aim:**

To write a python Program to implement case study for given scenario.

**Program:**

import random

hedges=("Please tell me more","Many of my patient tell me the same thing","Please continue ")

quantifiers=("why do you say that","you seem to think that","Can you explain why") replacements={"I":"you","me":"you","my":"your","we":"you","us":"you","mine":"yours"} def reply(sentence):

probablity=random.randint(1,4) if probablity==1:

return random.choice(hedges) else:

return random.choice(quantifiers)+changePerson(sentence) def changePerson(sentence):

words=sentence.split() replyWords=[]

for word in words: replyWords.append(replacements.get(word,word))

return "".join(replyWords) def main():

print("good morning, I hope well today") print("what can I do for you") sentence=input("\n>>")

while sentence.upper() !="QUIT": print(reply(sentence)) sentence=input("\n>>") print("have a nice day")

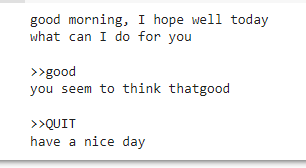
main()

**Screenshot:**

**Input:**

****

**Output:**

****

**Result:**

Thus the above program has been executed successfully.

**EX.No. : 19 RegNo:**

**Date : Name :**

**Program using classes and methods**

**Aim:**

To write the code for creating classes and methods.

**Program:**

class BankAccount:

def init (self, owner, balance=0): self.owner = owner

self.balance = balance

def deposit(self, amount): if amount > 0:

self.balance += amount print(f"Deposited ${amount}")

else:

print("Invalid deposit amount")

def withdraw(self, amount):

if amount > 0 and amount <= self.balance: self.balance -= amount print(f"Withdrew ${amount}")

else:

print("Invalid or insufficient funds")

def display\_balance(self):

print(f"{self.owner}'s account balance: ${self.balance}")

# Sample usage

account1 = BankAccount("Alice", 100)

account1.display\_balance() account1.deposit(50) account1.withdraw(30) account1.withdraw(150) account1.display\_balance()

**Output:**

Alice's account balance: $100 Deposited $50

Withdrew $30

Invalid or insufficient funds Alice's account balance: $120

**Result:**

Thus the above program has been executed successfully.

**EX.No. : 20 RegNo:**

**Date : Name :**

**Aim**

**Program using polymorphism, abstract classes**

To write a Python program that demonstrates polymorphism using abstract classes

and abstract methods, where different subclasses provide their own implementation of a common abstract method.

**Program**

from abc import ABC, abstractmethod # Abstract class

class Shape(ABC):

# Abstract method (must be implemented by subclasses) @abstractmethod

def draw(self): pass

# Concrete method def display(self):

print("Displaying shape details...")

# Subclass 1

class Circle(Shape): def draw(self):

print("Drawing a Circle")

# Subclass 2

class Rectangle(Shape): def draw(self):

print("Drawing a Rectangle")

# Subclass 3

class Triangle(Shape): def draw(self):

print("Drawing a Triangle")

# Main program def main():

# Polymorphism: the same method name works differently for each object shapes = [Circle(), Rectangle(), Triangle()]

for shape in shapes:

shape.draw() # Calls the subclass-specific method shape.display() # Calls the common method from abstract class print() # Blank line for clarity

if name == " main ": main()

**Output**

Drawing a Circle Displaying shape details...

Drawing a Rectangle Displaying shape details...

Drawing a Triangle Displaying shape details... **Result:**

Thus the above program has been executed successfully.